

AD-A115 782

NAVAL POSTGRADUATE SCHOOL MONTEREY CA
EFFICIENT MULTICHANNEL AUTOREGRESSIVE MODELING IN TIME AND FREQ--ETC(U)
MAR 82 D J KLICH

F/O 12/1

UNCLASSIFIED

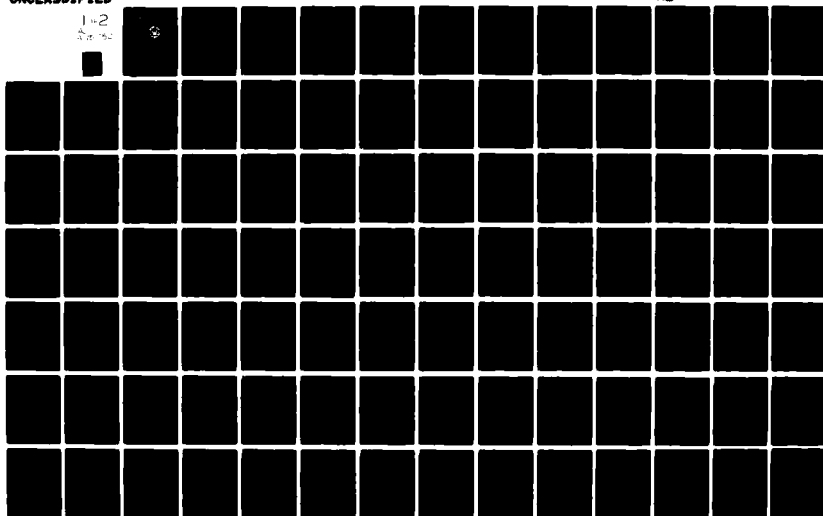
NL

1-2

3-4-5-6



7



AD A115752

2

NAVAL POSTGRADUATE SCHOOL

Monterey, California



DTIC

JUN 1982

THESIS

EFFICIENT MULTICHANNEL AUTOREGRESSIVE
MODELING IN TIME AND FREQUENCY DOMAIN

by

David J. Klich

March 1982

Thesis Advisor:

S. R. Parker

Approved for public release, distribution unlimited

DTIC FULL COPY

82 0 10 034

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Efficient Multichannel Autoregressive Modeling in the Frequency Domain		5. TYPE OF REPORT & PERIOD COVERED Dissertation March 1982
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) David J. Klich		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		12. REPORT DATE March 1982
		13. NUMBER OF PAGES 179
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release, distribution unlimited		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Autoregressive; Autoregressive Lattice; Autoregressive Moving Average; Autoregressive Moving Average Lattice; Autoregressive Moving Average Modeling; Autoregressive Modeling; System Identification; System Modeling		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) The single channel autoregressive lattice has been successfully applied to problems including speech analysis and recognition, spectral analysis and noise cancelling. More recently the two channel autoregressive (AR) lattice has been exploited for auto- regressive moving average (ARMA) analysis of systems for modeling and identification. This dissertation considers the multichannel AR lattice when applied to ARMA systems analysis. Constraints on lattice		

parameters, based on the input output relations of the system under test, are developed. The lattice is redefined in terms of the frequency domain representation of the input data. This proves to be useful because it allows the input to be normalized so that the lattice yeilds a consistant set of parameters independent of the test source characteristics. Lastly the lattice is redefined in terms of correlations of the input signals. This results in a computationally and storage efficient lattice algorithm.

Classification For		
DTIC	<input checked="" type="checkbox"/>	
AD	<input type="checkbox"/>	
AN	<input type="checkbox"/>	
Availability Codes		
Dist	Special	
A		



Approved for public release, distribution unlimited

Efficient Multichannel Autoregressive
Modeling In Time And Frequency Domain

by

David J. Klich
Lieutenant Commander, United States Navy
B.S., Virginia Polytechnic Institute, 1969

Submitted in partial fulfillment of the
requirements for the degree of

DOCTOR OF PHILOSOPHY

from the

NAVAL POSTGRADUATE SCHOOL
March 1982

Author: D. J. Klich

Approved:

U. R. Kodres

U. R. Kodres
Associate Professor of
Computer Science

K. G. Gray

K. G. Gray
Associate Professor of
Electrical Engineering

B. O. Shubert

B. O. Shubert
Associate Professor of
Operations Research,
Probability and Statistics

S. R. Parker

S. R. Parker
Professor of Electrical
Engineering
Dissertation Supervisor

G. A. Myers

G. A. Myers
Associate Professor of
Electrical Engineering

Approved by:

Robert D. Strum

Chairman, Department of Electrical Engineering

Approved by:

A. Schrey

Academic Dean

ABSTRACT

The single channel autoregressive lattice has been successfully applied to problems including speech analysis and recognition, spectral analysis and noise cancelling. More recently the two channel autoregressive (AR) lattice has been exploited for autoregressive moving average (ARMA) analysis of systems for modeling and identification.

This dissertation considers the multichannel AR lattice when applied to ARMA systems analysis. Constraints on lattice parameters, based on the input output relations of the system under test, are developed. The lattice is redefined in terms of the frequency domain representation of the input data. This proves to be useful because it allows the input to be normalized so that the lattice yields a consistent set of parameters independent of the test source characteristics. Lastly the lattice is redefined in terms of correlations of the input signals. This results in a computationally and storage efficient lattice algorithm.

TABLE OF CONTENTS

I.	INTRODUCTION-----	8
A.	THE PROBLEM STATEMENT-----	8
B.	OVERVIEW OF THE DISSERTATION-----	14
II.	LINEAR SYSTEM MODELING AND IDENTIFICATION-----	20
A.	THE AUTOREGRESSIVE MODEL-----	20
1.	A Recursive in Order Solution-----	25
2.	An AR Lattice Structure-----	30
B.	THE MOVING AVERAGE MODEL-----	37
1.	A Recursive in Order Solution-----	40
2.	A MA Lattice Structure-----	42
C.	MULTICHANNEL MODELING-----	44
1.	A Multichannel Generalization-----	44
III.	THE AUTOREGRESSIVE MOVING AVERAGE MODEL IN TIME AND FREQUENCY DOMAIN-----	54
A.	THE ARMA EQUATIONS-----	55
B.	A RECURSIVE IN ORDER ARMA SOLUTION-----	56
1.	An ARMA Lattice Structure-----	58
C.	LATTICE ALGORITHMS IN THE FREQUENCY DOMAIN---	62
1.	The Single Channel AR Lattice-----	64
2.	The Two Channel ARMA Lattice-----	68
D.	SYSTEM IDENTIFICATION WITH THE ARMA LATTICE--	72
1.	Restraints on ARMA Lattice Parameters----	73
2.	Uniqueness of the Lattice Parameters----	84
E.	FREQUENCY DOMAIN ARMA LATTICE SIMULATIONS----	92

F.	FILTER SYSTHESIS WITH THE ARMA FREQUENCY DOMAIN LATTICE-----	109
1.	Filter Synthesis Examples-----	112
IV.	AUTOREGRESSIVE LATTICE PARAMETERS CALCULATED USING THE CORRELATION MATRIX-----	121
A.	THE SINGLE CHANNEL AUTOREGRESSIVE LATTICE-----	122
1.	Implementation of the Single Channel Algorithm-----	134
B.	THE TWO CHANNEL LATTICE-----	137
1.	Correlation Lattice Simulations-----	142
V.	SUMMARY, CONSLUSIONS AND OPEN QUESTIONS-----	152
	APPENDIX A - THE TWO CHANNEL CORRELATION LATTICE-----	157
	LIST OF REFERENCES-----	176
	BIBLIOGRAPHY-----	178
	INITIAL DISTRIBUTION LIST-----	179

ACKNOWLEDGMENT

I must express my appreciation for the patient and committed guidance extended to me by Professor S. R. Parker. This project would have clearly been impossible without his continual support and insight.

I wish also to thank Professor Paul Moose who provided to me friendship and an outlet that provided stability in sometimes rough seas.

Most important, however, has been the understanding and forbearance of my wife Nancy. We did this together.

I. INTRODUCTION

In many applications the use of mathematical models provides critical insight into dynamic system behavior and is essential to many control, simulation and communication applications. Models have been applied in the areas of economics, neurophysics and astronomy with varying degrees of success. Engineers routinely use models when they wish to predict or control a system's performance.

In this dissertation one form of model, the multichannel lattice, is examined in detail in an effort to gain insight into its operation and to develop more efficient algorithms for its implementation and application to the modeling of multichannel linear and nonlinear autoregressive moving average (ARMA) systems, based upon a least mean square error approach.

A. THE PROBLEM STATEMENT

The output $y(k)$ of the discrete, linear, time invariant system of Figure 1.1 may be described as the weighted summation of the present and past values of the input $u(k)$ and past values of the output $y(k)$. This can be summarized by the relation

$$y(k) = \sum_{n=0}^q \alpha_n u(k-n) + \sum_{n=1}^p \beta_n y(k-n) \quad (1-1a)$$

defining the transfer function

$$\frac{Y(z)}{U(z)} = \frac{\alpha(z)}{\beta(z)}$$

where

$$\alpha(z) = \sum_{n=0}^q \alpha_n z^{-n} \quad \text{and} \quad \beta(z) = 1 - \sum_{n=1}^p \beta_n z^{-n} \quad (1-1b)$$

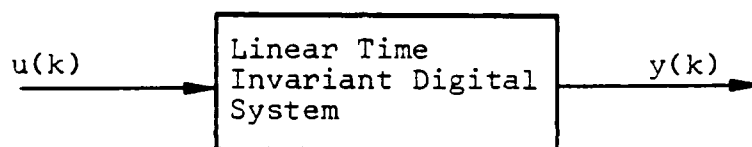


Figure 1.1 General Discrete System

where a_n , $0 \leq n \leq q$ and b_n , $1 \leq n \leq p$ are the parameters of the system and k denotes the sample index.

An estimate of the system parameters can be made if the output signal, $y(k)$, and possibly the input signal, $u(k)$, of a system can be observed. The result is a model of the system. There are numerous reasons why this might be done, among which are:

- 1) The parameters of the system are not known and some insight into the system function is desired.

- 2) The nominal values of the system parameters are known and we wish to detect a change in the system performance.

Equation (1-1) suggests that a system output is predictable from a linear combination of past outputs and inputs.

Because of this, procedures to find these parameters are often referred to as linear prediction.

One way to construct a model is to hypothesize a system of the same form, namely

$$\hat{y}(k) = \sum_{n=0}^q \hat{a}_n u(k-n) + \sum_{n=1}^p \hat{b}_n \hat{y}(k-n) \quad (1-2)$$

and adjust the model parameters \hat{a}_n and \hat{b}_n so that the difference between $y(k)$ and $\hat{y}(k)$ is minimized according to some criteria. Figure 1.2 shows this method. The symbol " $\hat{}$ " is used to indicate an estimated value. From Figure (1.2) and (1-1) and (1-2)

$$\begin{aligned} e(k) &= y(k) - \hat{y}(k) \\ &= \sum_{n=0}^q (\alpha_n - \hat{a}_n) u(k-n) + \sum_{n=1}^p \beta_n y(k-n) - \sum_{n=1}^k \hat{b}_n \hat{y}(k-n) \end{aligned}$$

Since $\hat{y}(k)$ is itself dependent on model parameters, and a minimum mean square output error criterion results in a set of nonlinear equations, the solution of this equation for model parameters is generally intractable when \hat{b}_n , $1 \leq n \leq p$ is not zero.

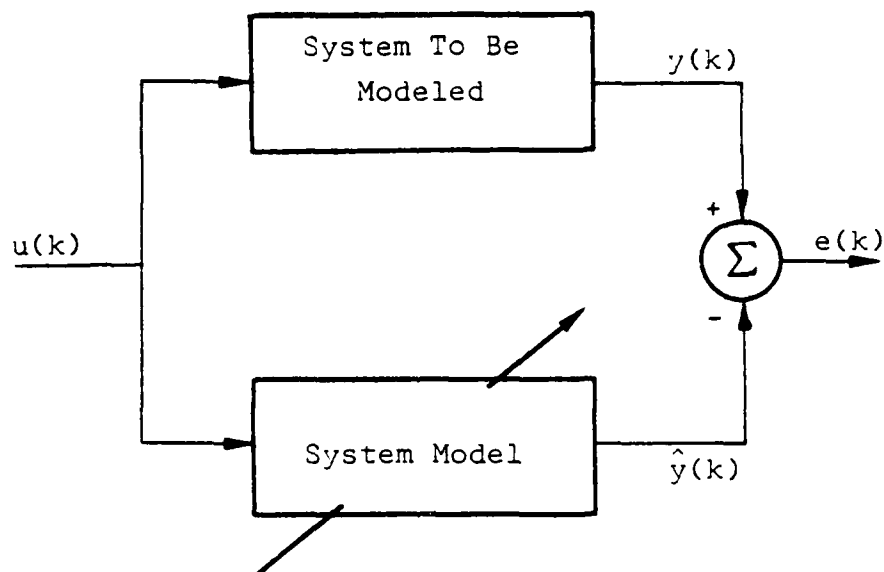


Figure 1.2 Direct System Modeling

Because of this, other analysis procedures have been developed. A very powerful model is the equation error formulation, which entails two models; the analysis model, with parameters a_n and b_n which are estimates of the system parameters, and a synthesis model using a_n and b_n to emulate the system.

The output of the equation error analysis model is written as

$$\hat{y}(k) = \sum_{n=0}^q a_n u(k-n) + \sum_{n=1}^p b_n y(k-n) \quad (1-3)$$

where the estimated output is now a function of the system past and present input and system past outputs (instead of the model outputs). This form is shown as Figure 1.3. With the equation error approach a MMSE solution for the model parameters is a set of simultaneous linear equations with a unique minimum. The synthesis model for the equation error approach is of the form of (1-2), using the estimated system parameters found from the analysis model. The equation error of Figure 1.3 is given by

$$e(k) = y(k) - \hat{y}(k) = \sum_{n=0}^q (\alpha_n - a_n)u(k-n) + \sum_{n=1}^p (\beta_n - b_n)y(k-n) \quad (1-4)$$

which is linear in the model parameters. When $\alpha_n = a_n$ and $\beta_n = b_n$, $e(k) = 0$. In terms of the z-transform, (1-4) can be written as

$$\begin{aligned} E(z) &= -A(z)U(z) + B(z)Y(z) \\ &= \left[-A(z) + \frac{B(z)}{\beta(z)} \alpha(z) \right] U(z) \end{aligned} \quad (1-5)$$

where

$$A(z) = \sum_{n=0}^q a_n z^{-n} \text{ and } B(z) = 1 - \sum_{n=1}^p b_n z^{-n}.$$

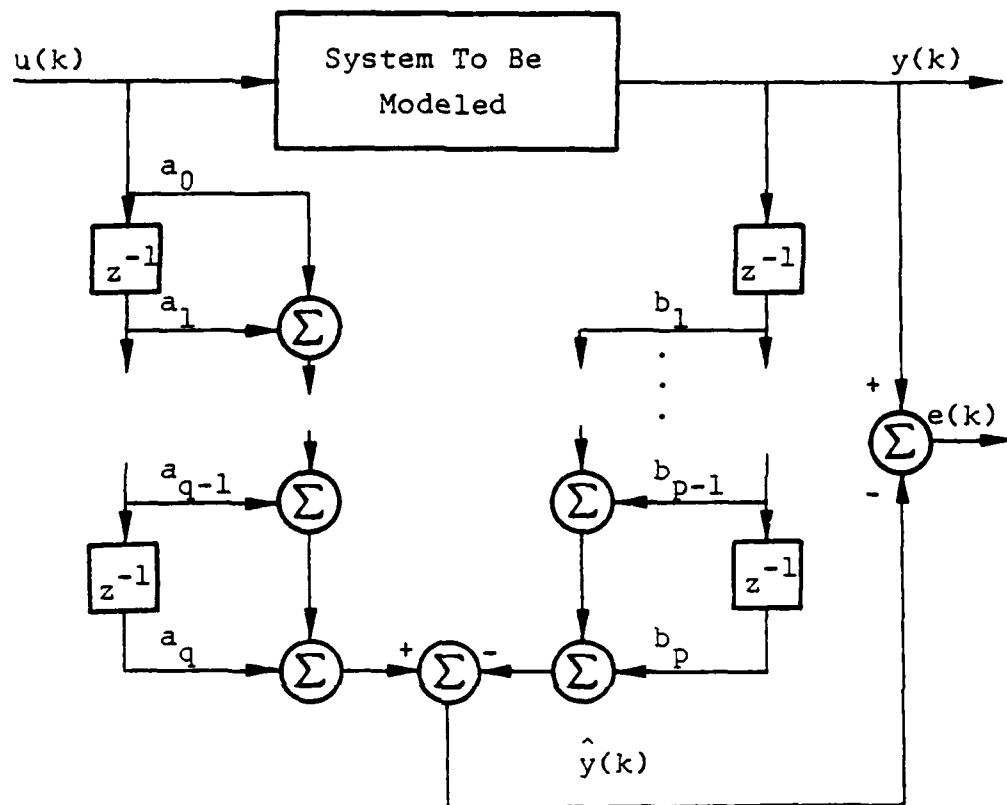


Figure 1.3 Equation Error Analysis Model

Three cases are examined in this dissertation.

- 1) The all pole, or autoregressive (AR), model
 $a_n = 0, 1 \leq n \leq q$
- 2) The all zero, or moving average (MA), model
 $b_n = 0, 1 \leq n \leq p$
- 3) The pole-zero, or autoregressive moving average (ARMA), model

$a_n \neq 0$, for some n , $1 \leq n \leq q$

$b_n \neq 0$, for some n , $1 \leq n \leq p$

The objective will be to develop procedures and algorithms that allow more efficient calculation of these models for application to system performance evaluation and identification.

B. OVERVIEW OF THE DISSERTATION

The following chapters are first a review of some contemporary thoughts on linear systems modeling using autoregressive, moving average and autoregressive moving average modeling, with their lattice formulation, followed by some new lattice configurations centered upon the ARMA lattice configurations.

Yule [19] first developed the autoregressive model in a paper on sunspot analysis. Because of its wide ranging applications, autoregressive modeling has received extensive examination since. Chapter II.A begins by reviewing the relevant theory of the least mean square solution of the AR equations. This solution results in a set of simultaneous equations that are the equivalent of the Yule-Walker [9] equations. Their resolution requires the solution of the matrix equation $\underline{R}\underline{b} = \underline{r}$, where \underline{R} is a symmetric and Toeplitz correlation matrix. A number of methods have been proposed for the solution of these equations. Matrix inversion methods can be used, but their computational complexity is of concern for higher order problems. Adaptive schemes [18]

have been developed that update an estimate of the solution as each data point is taken into account. Levinson [6] presented an elegant and efficient recursive method of performing this inversion by finding an $(n+1)$ -th order solution as an adjustment of the n -th order solution. Durbin [2] later recognized that the elements of the vector \underline{r} were also elements of the matrix \underline{R} . This was used to improve the computational efficiency of Levinson's algorithm. This recursive solution can be implemented as a lattice structure that has been found to offer an accurate and efficient solution to the AR modeling problem. More recently Burg [1] has shown that this lattice is a maximum entropy solution, proving that the lattice solution is optimum.

In Section II.B the moving average model is similarly discussed. The result of a least mean square solution is the digital form of the Weiner-Hopf [17] equations and also requires the inversion of a symmetric and Toeplitz matrix. Perry [16] has shown that this can be efficiently solved using Levinson's recursion and can also be restructured to form a lattice implementation, part of which is the AR model.

Sections II.A and II.B deal with single channel (single input, single output) models. Section II.C discusses multiple input, multiple output generalizations of the AR and MA equations and structures that Robinson [16] has shown can be solved as a multichannel generalization of the single channel structures.

Chapter III is a discussion of ARMA modeling. Unlike the AR and MA models, the ARMA model represents both poles and zeros and so will offer a lower order solution than either the AR or MA models for zero-pole systems. In III.A the least mean square solution of the ARMA problem is developed. A direct application of the ARMA equations results in a set of nonlinear, multimodal equations. This problem was greatly simplified when the equation error formulation [5, 12] was found to result in linear equations. Then in Section III.B it is shown how Perry [16] reconfigured, with appropriate assumptions, the ARMA problem so that it could be solved using a structure identical to the two channel AR lattice. This structure offers the advantages of a recursive in order solution and results in more accurate solutions than batch methods that use more direct matrix inversions.

Lattices have, in the past, used data in the form of a discrete time series as a block of data, as discussed in Chapter II and III.A and III.B. Griffiths [3] and later Morf [13] developed adaptive algorithms that still use discrete time series data but update an estimate of the lattice parameters as each data point enters the lattice structure. In Section III.C a new formulation for determining the lattice parameters is presented. This approach, although fundamental, has generally not been exploited, particularly for the multichannel case. In this structure

the data presented to the lattice is the frequency domain spectrum of the signal to be analysed. The power spectral density of the error spectrum is minimized and the resultant lattice is equivalent to that used with time domain data. This new lattice structure is useful because it allows the spectrum content of the input signals to be observed and presents the possibility of adjusting the input spectra to compensate for a nonwhite driving source. This configuration also permits the lattice to be used for filter synthesis.

Section III.D discusses the lattice ARMA model when applied to system identification. The ARMA lattice identifier can be used for system performance evaluation. In this application the model parameters obtained as a result of a test are compared with model parameters obtained from a system functioning normally and functioning with known faults. This comparison can reveal if the system under test is operating satisfactorily and possibly identify a class of faults within the system. The lattice parameters fully identify, in a least mean square sense, both the input and output of the system to be modeled. For modeling it is generally desirable that the input signal be white (have a flat spectrum) so that the signal energy is equally distributed across the spectrum of interest. Some new relations among the lattice parameters are developed, which, with some previously known relationships, reduce the number of parameters needed by one half when the input is properly

conditioned. Some new methods are then derived that allow preemphasis or normalization of the system input to correct for a nonwhite input signal. This is important because lattice parameters, while modeling the system, are still a function of the input signals second order statistics. This is useful even when the test source is purportedly white, since the lattice must function with estimates of signal parameters obtained from finite length data signals.

Section III.F introduces another application of the new frequency domain lattice. The lattice can be used to experimentally synthesize a filter to meet a desired specification by providing the desired response as the input of the lattice. The model generated is a minimum mean square error approximation of the desired specification.

In Chapter IV and Appendix A a new approach to both the single channel and two channel lattice is developed. This method recognizes that the error signals of the lattice are only linear combinations of the input signals and delayed versions of the input signals. From this observation the equations for the lattice parameters are restructured so that only linear combinations of the input correlation functions are needed to solve the lattice. This is more computational and storage efficient than other lattice methods since it is not now necessary to generate updated error signals at each stage. It retains the advantages of the lattice of orthogonality

and maximum entropy. It offers the new advantage that data need not be artificially windowed to force the correlation matrix to be Toeplitz, as is required by Levinson's algorithm. This may lead, with some types of data, to a more accurate solution. This method is extendable to higher dimensioned lattices as are used for nonlinear and multiple input multiple output system modeling.

Chapter V presents a summary of the new results in this dissertation. Included is a discussion of the application and implementation of these new concepts, along with questions for future research.

II. LINEAR SYSTEM MODELING AND IDENTIFICATION

The autoregressive, moving average, and autoregressive moving average models have all been successfully used for system modeling and analysis. The AR model is the model of choice when only system output is available. The MA model can give lower MSE if an input signal is available, but cannot generate an infinite impulse response. These models will be examined in greater detail and computationally efficient algorithms for their solution will be developed. Both single input and single output and multiple channel implementations are discussed.

A. THE AUTOREGRESSIVE MODEL

For the autoregressive model the assumption is made that the present output of a system can be estimated from a weighted summation of the past values of the output. This can be expressed as

$$\hat{y}(k) = \sum_{n=1}^p b_n y(k-n) \quad (2-1)$$

or in matrix form

$$\hat{y}(k) = \underline{b}^T \underline{y}(k) \quad (2-2)$$

with

$$\underline{b}^T = [b_1 b_2 \dots b_p]$$

$$\underline{y}^T(k) = [y(k-1) \ y(k-2) \ \dots \ y(k-p)]$$

If a signal $y(k)$ is to be modeled the prediction error can be written as the difference in the signal and its estimate or

$$e(k) = y(k) - \hat{y}(k) = y(k) - \underline{b}^T \underline{y}(k) \quad (2-3)$$

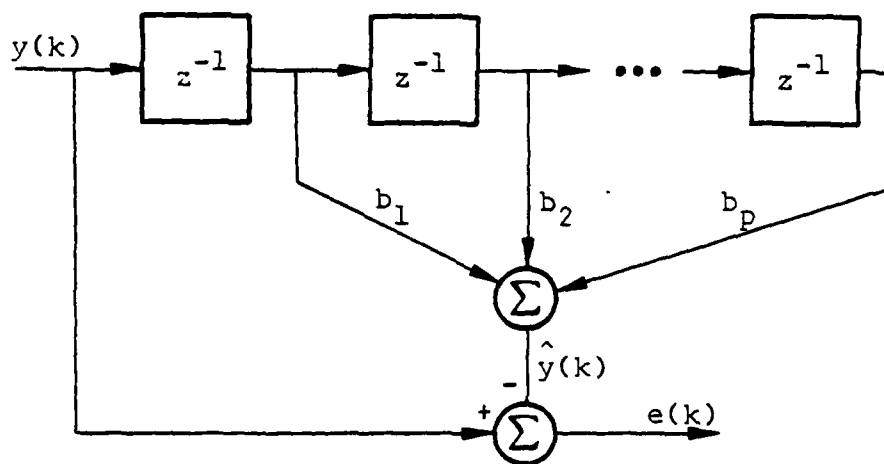


Figure 2.1 An AR Prediction Error Model

Figure 2.1 illustrates this type of system.

The mean squared prediction error as a function of the weights b_n is found to be

$$E = \underline{b}^T \underline{R}_{yy} \underline{b} - 2 \underline{b}^T \underline{r}_{yy} + R_{yy}(0) \quad (2-4)$$

where in general $R_{xw}(n) = \epsilon\{x(k)w(k+n)\}$, $\underline{r}_{xw} = \epsilon\{\underline{x}(k)w(k+n)\}$, $\underline{R}_{xw} = \epsilon\{\underline{x}(k)\underline{w}^T(k)\}$ and $\epsilon\{\cdot\}$ designates expectation.

In this case,

$$\underline{R}_{yy} = \begin{bmatrix} R_{yy}(0) & \dots & R_{yy}(-p+1) \\ \vdots & & \vdots \\ R_{yy}(p-1) & \dots & R_{yy}(0) \end{bmatrix}$$

$$\underline{r}_{yy}^T = [R_{yy}(1) \dots R_{yy}(p)]$$

The solution of (2-4) for the optimum weights is given by

$$\underline{b}_{opt} = \underline{R}_{yy}^{-1} \underline{r}_{yy} \quad (2-5)$$

which are commonly called the normal equations. If this is substituted into (2-4) then the minimum error is found to be

$$E_{min} = R_{yy}(0) - \underline{b}_{opt}^T \underline{r}_{yy} \quad (2-6)$$

(2-3) may be written in the z domain as

$$\frac{E(z)}{Y(z)} = 1 - \sum_{n=1}^p b_n z^{-n} = B(z) \quad (2-7)$$

which is the relationship of error signal output to the input signal.

Consider the all pole filter of Figure 2.2 with the input signal $u(k)$ and output $y(k)$. The relationship between input and output can be written in the z domain

$$\frac{Y(z)}{U(z)} = H(z) = \frac{a_0}{1 - \sum_{n=1}^p b_n z^{-n}} = \frac{a_0}{B(z)} \quad (2-8a)$$

or in the time domain

$$y(k) = a_0 u(k) + \sum_{n=1}^p b_n y(k-n) \quad (2-8b)$$

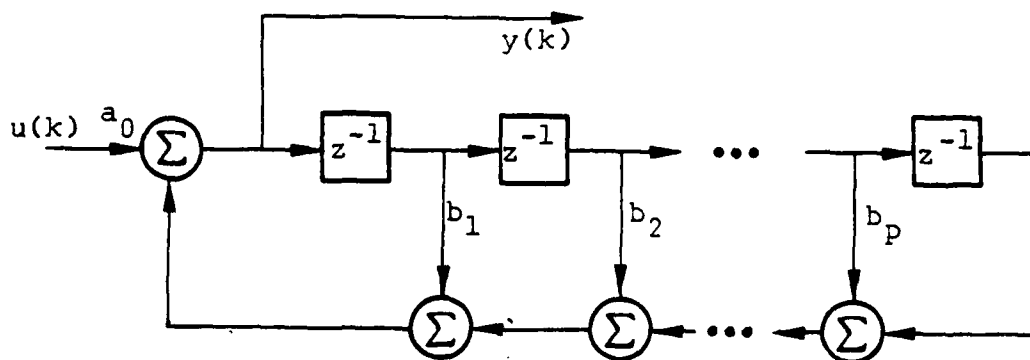


Figure 2.2 All Pole Filter

If the output $y(k)$ of the all pole filter is applied to the input of the prediction error model of Figure 2.1 the output error will be $e(k) = a_0 u(k)$. The transfer function of this cascade arrangement from the input of the all pole filter to the error output of the predictor will be a_0 . This shows that if $u(k)$ is white noise the effect of the prediction error

model is, except for a gain term a_0 , to reverse the action of the all pole filter or to whiten it's output. For this reason the prediction error model is often referred to as a whitening filter or an inverse filter, and makes it useful as a spectrum analyzer. It should be noted that a_0 establishes a lower bound on the MSE of the AR predictor. If $u(k)$ is a unit impulse, $u(k) = \delta(k)$, the model may replicate the response $y(k)$ perfectly, but can never predict the arrival of the input $a_0\delta(k)$ since the predictor is causal, and there is always this minimal error.

The relationship between an all pole filter of Figure 2.2 and the analysis model of (2-1) suggests that the all pole filter can be used as a synthesis model for generating the signal $y(k)$ using $a_0u(k)$ as the input signal with the coefficients obtained from the prediction error analysis model. It is only necessary to find a suitable gain term a_0 . Equation (2-3) can be rewritten

$$y(k) = e(k) + \underline{b}^T \underline{y}(k) \quad (2-9)$$

Comparing this with (2-8b) it is seen that $a_0u(k) = e(k)$. By requiring that the total energy of the output from the linear predictor be equal to the total energy of the input to the all pole filter, a_0 is specified by

$$a_0^2 = \frac{\epsilon\{e^2(k)\}}{R_{uu}(0)} \quad (2-10)$$

This leads to a z domain transfer function representation for the synthesis model of

$$H(z) = \frac{a_0}{B(z)} \quad (2-11)$$

The impulse response of (2-11) can be of infinite duration. This means that a relatively low order AR model may emulate an infinite impulse response system to an acceptable accuracy.

In the problem as stated so far it has been tacitly assumed that the desired order for the model was somehow known. In practice this is often not the case. Instead, (2-5) is solved for some order and the MSE determined. If this MSE is unacceptable, then (2-5) is again solved for a higher order. Solution of the simultaneous equations of (2-5) requires, by Gaussian elimination, $p^3/3 + O(p^2)$ operations, so that a repetitive solution of the normal equations can become computationally ponderous. There is, however, a high degree of symmetry in these equations and the exploitation of this symmetry can give a more efficient solution of (2-5).

1. A Recursive In Order Solution

The normal equations (2-5) for the optimum weights can be written in matrix form

$$\begin{bmatrix} R_{yy}(0) & R_{yy}(-1) & \dots & R_{yy}(-p+1) \\ R_{yy}(1) & R_{yy}(0) & & \\ \vdots & & \ddots & \vdots \\ R_{yy}(p-1) & \dots & & R_{yy}(0) \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_p \end{bmatrix} = \begin{bmatrix} R_{yy}(1) \\ R_{yy}(2) \\ \vdots \\ R_{yy}(p) \end{bmatrix} \quad (2-12)$$

The autocorrelation matrix is Toeplitz (the elements along any diagonal are equal) if the signals are stationary which is a basic assumption of this work.

Calculation of the autocorrelation functions now deserves some comment. Computation of the exact discrete correlation requires a summation of infinite duration signals.

$$R_{yy}(N) = \sum_{k=-\infty}^{\infty} y(k)y(k+N)$$

The result is an even function, or

$$R_{yy}(N) = R_{yy}(-N)$$

so that the autocorrelation matrix is symmetric and Toeplitz. In the absence of an infinite duration signal, the more common case, an estimate of the correlation function must be made. If a window function is used and the data is taken to be zero outside the window then the estimated correlation matrix will be symmetric when the data is averaged over the width of the window. If the data is

not taken to be zero outside the window and averaging is done over a finite number of points, the estimated correlation matrix is in general not symmetric unless a large number of data points are taken because of end effects. In the derivations which follow symmetric Toeplitz matrices are assumed. It is noted that the final algorithms can be used without assuming symmetric Toeplitz correlation matrices, and apparently work just as well. In fact it is noted that Griffith's time adaptive technique [3] is nonstationary and yet appears to work very well. This has not been explained in the literature. If the observed signal is ergodic an estimation of the correlation function can be made using a finite interval of data.

When the data is windowed so that the autocorrelation matrix is Toeplitz and symmetric, the method used to solve the resultant normal equations is often referred to in the literature [9] as an autocorrelation method of solution. When the data is not so windowed and the matrix is not Toeplitz, the problem solution is said to be a covariance method. Either method results in an estimate of the true values of the correlation matrix, and it is generally unclear that one method will be more accurate than the other.

The Levinson algorithm relates the $(n+1)$ -th order solution of a set of equations of the form of (2-12) to the n -th order solution. The assumption is made that $\underline{b}^{(n+1)}$,

where the superscript denotes the order of the solution, can be found from $\underline{b}^{(n)}$ by adding a correction to each term and one new term, or

$$\underline{b}^{(n+1)} = \begin{bmatrix} \underline{b}^{(n)} \\ \hline 0 \end{bmatrix} + \begin{bmatrix} \xi^{(n)} \\ \hline k^{(n+1)} \end{bmatrix} \quad \text{where } \underline{b}^{(n)} = \begin{bmatrix} b_1^{(n)} \\ \vdots \\ b_n^{(n)} \end{bmatrix} \quad (2-13)$$

To find $\underline{b}^{(n+1)}$ parameters $\xi^{(n)}$ and $k^{(n+1)}$ must be computed. Permuted versions of the vectors $\underline{b}^{(n)}$ and $\underline{r}_{yy}^{(n)}$ may be defined by reversing the order of their elements.

$$\tilde{\underline{b}}^{(n)} = \begin{bmatrix} b_n^{(n)} \\ \vdots \\ b_1^{(n)} \end{bmatrix} \quad \underline{\rho}_{yy}^{(n)} = \begin{bmatrix} R_{yy}^{(n)} \\ \vdots \\ R_{yy}^{(1)} \end{bmatrix} = \tilde{\underline{r}}_{yy}^{(n)} \quad (2-14)$$

The symmetry of the autocorrelation matrix allows (2-12) with $p=n$ namely $\underline{R}_{yy}^{(n)} \underline{b}^{(n)} = \underline{r}_{yy}^{(n)}$, to be rewritten using (2-14) as

$$\underline{R}_{yy}^{(n)} \tilde{\underline{b}}^{(n)} = \underline{\rho}_{yy}^{(n)} \quad (2-15)$$

The matrix $\underline{R}_{yy}^{(n+1)}$ is recognized as the matrix $\underline{R}_{yy}^{(n)}$ with one new row and column and may be written as

$$\underline{R}_{yy}^{(n+1)} = \begin{bmatrix} & & R_{yy}^{(-n)} \\ & & \vdots \\ & & R_{yy}^{(-1)} \\ \hline R_{yy}^{(n)} & & \\ \vdots & & \\ R_{yy}^{(n)} \dots R_{yy}^{(1)} & R_{yy}^{(0)} \end{bmatrix} = \begin{bmatrix} R_{yy}^{(n)} & \underline{r}_{yy}^{(n)} \\ \hline \underline{\rho}_{yy}^{(n)T} & R_{yy}^{(0)} \end{bmatrix}$$

since the data has been properly windowed to ensure that the autocorrelation function estimate is an even function of time. The foregoing and (2-13) may be substituted into an (n+1)-th order version of (2-12) to give

$$\begin{bmatrix} \underline{R}_{yy}^{(n)} & | & \underline{\rho}_{yy}^{(n)} \\ \hline \underline{\rho}_{yy}^{(n)T} & | & R_{yy}^{(0)} \end{bmatrix} \begin{bmatrix} \underline{b}^{(n)} + \underline{\xi}^{(n)} \\ \hline k^{(n+1)} \end{bmatrix} = \begin{bmatrix} \underline{r}_{yy}^{(n)} \\ \hline R_{yy}^{(n+1)} \end{bmatrix}$$

Multiplying the matrices and substituting (2-15) yields

$$\underline{\xi}^{(n)} = -\underline{b}^{(n)} k^{(n+1)} \quad (2-16a)$$

where

$$k^{(n+1)} = \frac{R_{yy}^{(n+1)} - \underline{\rho}_{yy}^{(n)T} \underline{b}^{(n)}}{R_{yy}^{(0)} - \underline{b}^{(n)T} \underline{r}_{yy}^{(n)}} \quad (2-16b)$$

Substituting (2-16a) into (2-13) yields

$$\underline{b}^{(n+1)} = \begin{bmatrix} \underline{b}^{(n)} \\ \hline 0 \end{bmatrix} + k^{(n+1)} \begin{bmatrix} -\underline{b}^{(n)} \\ \hline 1 \end{bmatrix} \quad (2-17)$$

To find $\underline{b}^{(n+1)}$ from $\underline{b}^{(n)}$ it is only necessary to determine $k^{(n+1)}$, which can be found from (2-16b) which uses $R_{yy}^{(n+1)}$ and other terms from the previous solution. It is noted by comparison with (2-6) that the denominator of (2-17)

is the MMSE of the (n)-th order solution for the normal equations. Equations (2-13), (2-14), (2-16) and (2-17) can be applied recursively, starting with order 1, until a model results with the desired accuracy.

2. An AR Lattice Structure

The vector $\underline{b}^{(n)}$ can be written in the transform domain

$$B^{(n)}(z) = 1 - \sum_{i=1}^n b_i^{(n)} z^{-i} = 1 - b_1^{(n)} z^{-1} - \dots - b_n^{(n)} z^{-n} \quad (2-18)$$

and the reversed in order version $\tilde{\underline{b}}^{(n)}$ can be written

$$\bar{B}^{(n)}(z) = z^{-n} B^{(n)}(z^{-1}) \quad (2-19)$$

It should be recalled that $B(z)$ is the transfer function of the prediction error model. The overbar here is used to indicate the reversal in order, and will be used throughout this dissertation to signify a backward signal, or a function or matrix associated with a backward signal.

A recursive solution for the transfer function $B(z)$ can be found by substituting (2-17) into an (n+1) order version of (2-18)

$$B^{(n+1)}(z) = 1 - [z^{-1} z^{-2} \dots z^{-n-1}] \underline{b}^{(n+1)}$$

$$= 1 - [z^{-1}z^{-2}\dots z^{-n-1}] \begin{bmatrix} b^{(n)} \\ \hline 0 \end{bmatrix} + k^{(n+1)} \begin{bmatrix} -\tilde{b}^{(n)} \\ \hline 1 \end{bmatrix}$$

$$= B^{(n)}(z) - z^{-1}k^{(n+1)}z^{-n}[z^n z^{n-1} \dots 1] \begin{bmatrix} -\tilde{b}^{(n)} \\ \hline 1 \end{bmatrix}$$

and substituting (2-19) into the second term of the foregoing.

The result is

$$B^{(n+1)}(z) = B^{(n)}(z) - z^{-1}k^{(n+1)}\bar{B}^{(n)}(z) \quad (2-20)$$

$\bar{B}^{(n+1)}(z)$ can also be found recursively by substituting (2-20) into (2-19) rewritten for order (n+1)

$$\bar{B}^{(n+1)}(z) = z^{-1}\bar{B}^{(n)}(z) - k^{(n+1)}B^{(n)}(z) \quad (2-21)$$

Since $B^{(n)}(z)$ is the transfer function of the n-th order prediction error model, when it is driven by the signal $Y(z)$ the result is the n-th order error signal, or

$$E^{(n)}(z) = B^{(n)}(z)Y(z) \quad (2-22)$$

The forward prediction error is the difference in the signal and a predicted signal derived from a weighted summation of the past n samples of the signal. The backward prediction

error is the difference in the signal at time $(k-n)$ and its predicted value based on a weighted summation of the n samples to follow. Figure 2.3 is an illustration of this forward and backward prediction.

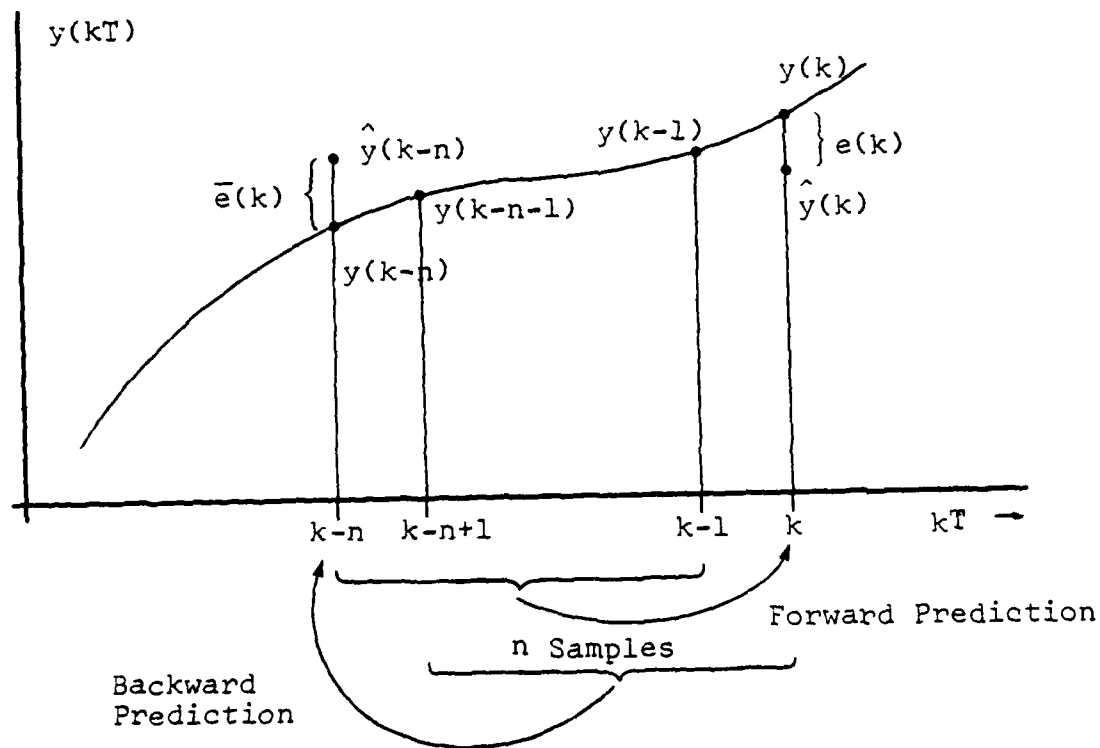


Figure 2.3 Illustration of Forward and Backward Prediction Using n Points

By multiplying (2-20) and (2-21) by $Y(z)$, $E(z)$ and $\bar{E}(z)$ may also be found recursively.

$$E^{(n+1)}(z) = E^{(n)}(z) - k^{(n+1)} z^{-1} \bar{E}^{(n)}(z) \quad (2-23)$$

$$\bar{E}^{(n+1)}(z) = z^{-1} \bar{E}^{(n)}(z) - k^{(n+1)} E^{(n)}(z) \quad (2-23a)$$

In the time domain these two equations become

$$e^{(n+1)}(k) = e^{(n)}(k) - k^{(n+1)}\bar{e}^{(n)}(k-1) \quad (2-24)$$

$$\bar{e}^{(n+1)}(k) = \bar{e}^{(n)}(k-1) - k^{(n+1)}e^{(n)}(k) \quad (2-25)$$

Since the zero order prediction of a signal is $\hat{y}(k) = 0$ (no prediction at all) the zero order error is

$$e^{(0)}(k) = \bar{e}^{(0)}(k) = y(k) \quad (2-26)$$

and equations (2-24), (2-25) and (2-26) can be implemented as a lattice structure as is shown in Figure 2.4.

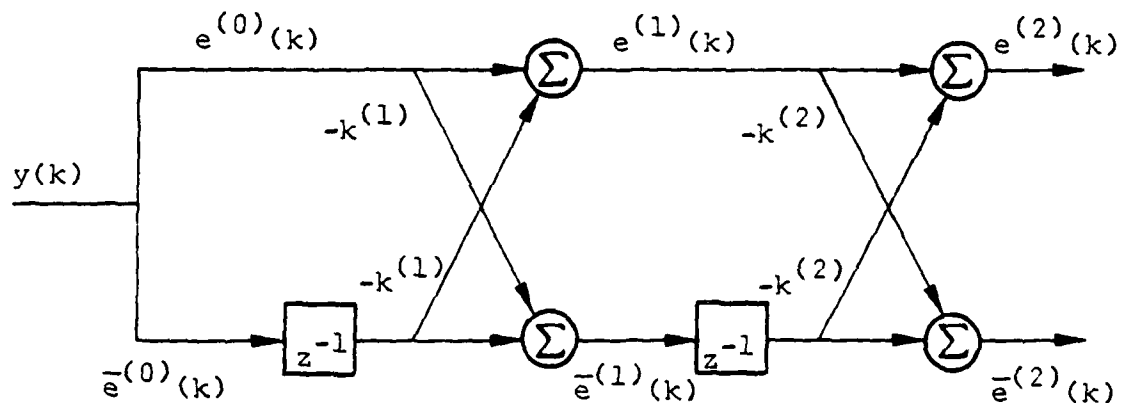


Figure 2.4 Second Order AR Lattice Model

The lattice parameter $k^{(n+1)}$ can be found from (2-16b) (as was done for the Levinson algorithm) or can be found by minimizing the expected value of the squared error of (2-24) or (2-25). Minimizing square of the the expected value of (2-24) with respect to k yields

$$k_F^{(n+1)} = \frac{\varepsilon\{e^{(n)}(k)\bar{e}^{(n)}(k-1)\}}{\varepsilon\{\bar{e}^{(n)}(k-1)\bar{e}^{(n)}(k-1)\}} \quad (2-27)$$

and similarly for (2-25)

$$k_B^{(n+1)} = \frac{\varepsilon\{e^{(n)}(k)\bar{e}^{(n)}(k-1)\}}{\varepsilon\{e^{(n)}(k)e^{(n)}(k)\}} \quad (2-28)$$

These alternative solutions are referred to as the forward and backward methods and can be shown [8] to be equivalent to (2-16b). From (2-20) and (2-21) it is seen that the forward and backward prediction transfer functions are equal so (2-27) and (2-28) must be equivalent. However, when implemented, the expectations will be estimated from time averages and will generally not be equal. This has led to two alternative solutions, the first a harmonic mean of (2-20) and (2-21) due to Burg [1]

$$k_{\text{BURG}}^{(n+1)} = \frac{\varepsilon \{e^{(n)}(k) \bar{e}^{(n)}(k-1)\}}{\frac{1}{2} \varepsilon \{e^{(n)2}(k) + \bar{e}^{(n)2}(k-1)\}} \quad (2-29)$$

and another from Itakaura and Saito [4] which is the geometric mean of (2-27) and (2-28)

$$k_{\text{I.S.}}^{(n+1)} = \frac{\varepsilon \{e^{(n)}(k) \bar{e}^{(n)}(k-1)\}}{\sqrt{\varepsilon \{e^{(n)2}(k)\} \varepsilon \{\bar{e}^{(n)2}(k-1)\}}} \quad (2-30)$$

Stability of the result is ensured if all the poles of the transform domain transfer function lie within the unit circle. Markel and Gray [11] have shown that the necessary and sufficient condition for this is

$$|k^{(n)}| < 1 \quad (2-31)$$

Since (2-30) is of the form of a normalized correlation it is of necessity always bounded by unity, and since the magnitude of a geometric mean is an upper bound of the magnitude of a harmonic mean, (2-30) and (2-29) are guaranteed to result in stable solutions. There is no such guarantee with the forward (2-27) and backward (2-28) methods.

A most important property of the lattice structure is the orthogonality of the backward error signals, described by

$$\epsilon \{e^{(i)}(k) \bar{e}^{(j)}(k)\} = \begin{cases} E_i, & i=j \\ 0, & i \neq j \end{cases} \quad (2-32)$$

Indeed, the lattice structure can be derived [11] starting with this property. This orthogonality decouples successive stages of the lattice, ensuring that the addition of stages in the recursive solution of the filter will not affect preceding stages, and permitting the derivation of (2-27) and (2-28). Furthermore, [8] orthogonality allows for a recursive calculation of the error energy at successive stages by

$$\epsilon \{e^{(n+1)}(k) e^{(n+1)}(k)\} = \epsilon \{\bar{e}^{(n+1)}(k) \bar{e}^{(n+1)}(k)\} = p^{(n+1)}$$

$$p^{(n+1)} = p^{(n)} (1 - k^{(n+1)^2}) \quad (2-33)$$

Noting that

$$p^{(0)} = \epsilon \{e^{(0)}(k) e^{(0)}(k)\} = R_{yy}(0) \quad (2-34)$$

(2-33) and (2-34) may be used for the denominator of either (2-16b), (2-27) or (2-28).

Once (2-17) is solved, the vector $\underline{b}^{(n)}$ may be implemented as the all pole filter of Figure 2.2, or other equivalent arrangements, to produce a synthesis model. The lattice synthesis structure may be generated by (2-25) and rearranging (2-24).

$$e^{(n)}(k) = e^{(n+1)}(k) + k^{(n+1)} \bar{e}^{(n)}(k-1) \quad (2-35)$$

This equation can be realized by the structure of Figure 2.5 and implements the transfer function $1/B^{(n)}(z)$.

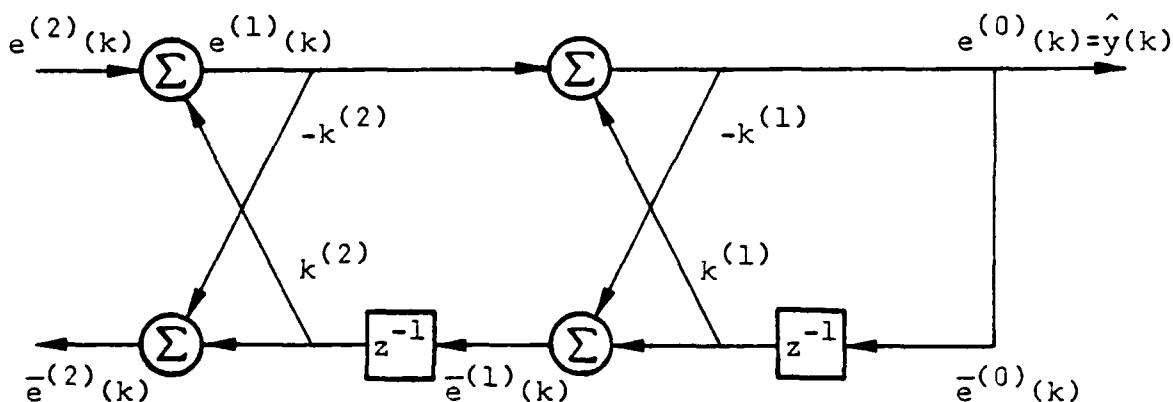


Figure 2.5 Second Order AR Lattice Synthesis Structure

B. THE MOVING AVERAGE MODEL

The moving average model is based on the assumption that the present value of a system output can be estimated by a weighted summation of the present and past q values of the system input. In equation form this is

$$\hat{y}(k) = \sum_{n=0}^q a(n) u(k-n) = \underline{a}^T \underline{u}(k) \quad (2-36)$$

where

$$\underline{a}^T = [a(0) \ a(1) \ \dots \ a(q)]$$

$$\underline{u}^T(k) = [u(k) \ u(k-1) \ \dots \ u(k-q)] \quad (2-37)$$

It should be noted that the vectors of (2-37) are (q+1) elements long.

Defining the error of the model as the difference in the estimated value and the actual value of the system output, or

$$e(k) = y(k) - \hat{y}(k) \quad (2-38)$$

the mean squared error E is found to be

$$E = \underline{a}^T \underline{R}_{uu} \underline{a} - 2 \underline{a}^T \underline{r}_{uy} + R_{yy}(0) \quad (2-39)$$

where \underline{R}_{uu} is (q+1)x(q+1) and \underline{r}_{uy} is a (q+1) element vector. When this is minimized with respect to the coefficients of \underline{a} the result is the normal equations of the moving average problem

$$\underline{R}_{uu} \underline{a} = \underline{r}_{uy} \quad (2-40)$$

From this the optimum value for \underline{a} is

$$\underline{a}_{\text{opt}} = \underline{R}_{uu}^{-1} \underline{r}_{uy} \quad (2-41)$$

while the MMSE is found by substituting (2-41) into (2-39)

$$E_{\text{min}} = R_{yy}(0) - \underline{a}_{\text{opt}}^T \underline{r}_{uy} \quad (2-42)$$

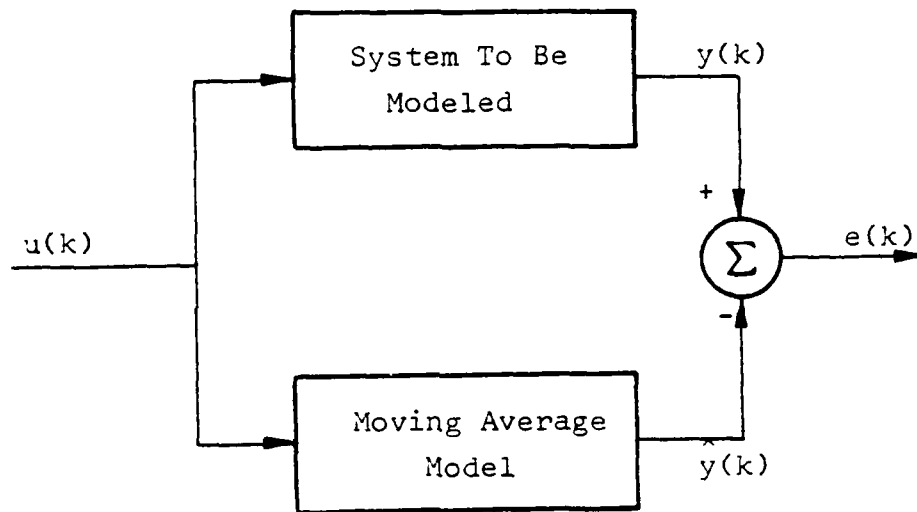


Figure 2.6 Moving Average Modeling

Figure 2.6 shows the block diagram of the implementation of the MA model. The transfer function of the model is, in the z domain

$$A(z) = \sum_{n=0}^q a(n)z^{-n} \quad (2-43)$$

and is often referred to as an all zero model. As such its impulse response is finite and in all cases the model is stable.

1. A Recursive In Order Solution

Equation (2-40) is similar in structure to (2-12).

R_{uu} is Toeplitz as is R_{yy} of (2-12), but the elements of r_{uy} of (2-40) are not also elements of the matrix R_{uu} . Still Levinson's algorithm may be applied.

Equation (2-40) is written for order n as

$$R_{uu}^{(n)} \underline{a}^{(n)} = \underline{r}_{uy}^{(n)} \quad (2-44)$$

and the assumption is made that $\underline{a}^{(n+1)}$ is a function of $\underline{a}^{(n)}$ and a correction vector.

$$\underline{a}^{(n+1)} = \begin{bmatrix} \underline{a}^{(n)} \\ \text{---} \\ 0 \end{bmatrix} + \begin{bmatrix} \underline{y}^{(n+1)} \\ \text{---} \\ \underline{g}^{(n+1)} \end{bmatrix} \quad (2-45)$$

Define vectors

$$\underline{r}_{uu}^{(n+1)} = \begin{bmatrix} R_{uu}(1) \\ \vdots \\ R_{uu}(n+1) \end{bmatrix}, \quad \underline{p}_{uu}^{(n+1)} = \begin{bmatrix} R_{uu}(n+1) \\ \vdots \\ R_{uu}(1) \end{bmatrix} = \tilde{\underline{r}}_{uu}^{(n+1)} \quad (2-46)$$

and expand an (n+1) order version of (2-44) with (2-45) substituted.

$$\begin{bmatrix} \underline{R}_{uu}^{(n)} & | & \underline{\rho}_{uu}^{(n+1)} \\ \hline \underline{\rho}_{uu}^{(n+1)T} & | & R_{uu}(0) \end{bmatrix} \begin{bmatrix} \underline{a}^{(n)} + \underline{\gamma}^{(n+1)} \\ \hline \underline{g}^{(n+1)} \end{bmatrix} = \begin{bmatrix} \underline{r}_{uy}^{(n)} \\ \hline R_{uy}^{(n+1)} \end{bmatrix}$$

When this is multiplied two equations result

$$\underline{R}_{uu}^{(n)} \underline{a}^{(n)} + \underline{R}_{uu}^{(n)} \underline{\gamma}^{(n+1)} + \underline{\rho}_{uu}^{(n+1)} \underline{g}^{(n+1)} = \underline{r}_{uy}^{(n)} \quad (2-47)$$

$$\underline{\rho}_{uu}^{(n+1)T} \underline{a}^{(n)} + \underline{\rho}_{uu}^{(n+1)T} \underline{\gamma}^{(n+1)} + R_{uu}(0) \underline{g}^{(n+1)} = R_{uy}^{(n+1)} \quad (2-48)$$

The first and last terms of (2-47) are equal so both may be cancelled leaving, when solved for $\underline{\gamma}^{(n+1)}$

$$\underline{\gamma}^{(n+1)} = -\underline{R}_{uu}^{(n)-1} \underline{\rho}_{uu}^{(n+1)} \underline{g}^{(n+1)} \quad (2-49)$$

Finding $\underline{R}_{uu}^{(n)-1} \underline{\gamma}_{uu}^{(n)}$ is comparable to (2-5), the (n+1)-st order autoregression of the input signal $u(k)$. So to derive the MA model, the AR model of the input must first be solved. When this is done a new vector $\underline{f}^{(n+1)}$ can be defined where

$$\underline{f}^{(n+1)} = \underline{R}_{uu}^{(n)-1} \underline{p}_{uu}^{(n+1)} = \begin{bmatrix} b_{n+1}^{(n+1)} \\ \vdots \\ b_1^{(n+1)} \end{bmatrix} \quad (2-50)$$

is the result of the autocorrelation analysis. Now $\underline{y}^{(n+1)}$ is defined as

$$\underline{y}^{(n+1)} = - \underline{f}^{(n+1)} g^{(n+1)} \quad (2-51)$$

and this can be substituted into (2-48) and solved for $g^{(n+1)}$.

$$g^{(n+1)} = \frac{R_{uy}^{(n+1)} - \underline{p}_{uu}^{(n+1)T} \underline{a}^{(n)}}{R_{uu}^{(n+1)} - \underline{p}_{uu}^{(n+1)T} \underline{f}^{(n+1)}} \quad (2-52)$$

2. A MA Lattice Structure

The Levinson solution of (2-49) can be restructured to give a lattice configuration. The transfer function of the MA model (2-43) can be written recursively in the z domain using (2-45) and (2-51) as

$$A^{(n+1)}(z) = A^{(n)}(z) + g^{(n+1)} \bar{B}^{(n+1)}(z) \quad (2-53)$$

where $\bar{B}^{(n+1)}(z)$ is the result of the autoregressive analysis of the input $u(k)$. If both sides of (2-53) are multiplied by the input $U(z)$ and then transformed to the time domain the result is the estimate of the output signal $y(k)$ for

$$\hat{y}^{(n+1)}(k) = \hat{y}^{(n)}(k) + g^{(n+1)}\bar{e}^{(n+1)}(k) \quad (2-54)$$

where $\bar{e}^{(n+1)}(k)$ is the backward error signal of the autoregressive analysis of $u(k)$. The desired lattice structure is shown in Figure 2.7.

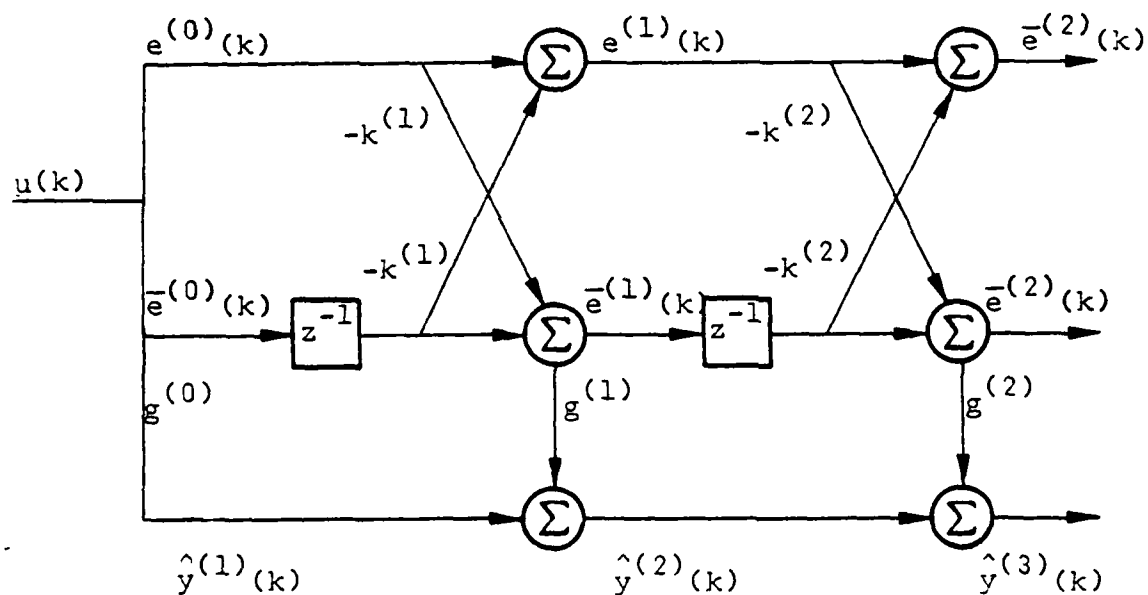


Figure 2.7 A Second Order MA Lattice Structure

Recalling that the backward error signals of different stages of an AR lattice are orthogonal, it is seen that the MA estimate of $y^{(n)}(k)$ is a linear weighted sum of these orthogonal signals.

C. MULTICHANNEL MODELING

The models previously considered have been of single input, single output systems. Multiple input, multiple output systems can be modeled as vector generalizations of the models heretofore developed.

1. A Multichannel Generalization

A single channel output of a Q-channel autoregressive model is the weighted summation of the past outputs of the Q channels of the system to be modeled. Similarly, a single channel output of a Q-channel moving average model of a system is the weighted summation of the past N inputs of the Q channels to the system to be modeled. Either of these statements may be written in equation form

$$\hat{x}_j(k) = \sum_{i=1}^Q \sum_{n=1}^N d_{ij}(n) x_i(k-n) \quad (2-55)$$

Q = number of channels

k = time index

n = time delay index

j = output channel being
estimated

i = input channel

$\hat{x}_j(k)$ = estimate of output channel j at time k

$x_i(k-n)$ = input to channel i at time $(k-n)$

$d_{ij}(n)$ = weighting factor for estimating the output signal at channel j at time k due to input $x_i(k-n)$

As an example, for a model with 3 inputs ($Q = 3$),
(2-55) would be expanded as

$$\hat{x}_1(k) = \sum_{n=1}^N d_{11}(n) x_1(k-n) + \sum_{n=1}^N d_{21}(n) x_2(k-n) + \sum_{n=1}^N d_{31}(n) x_3(k-n)$$

$$\hat{x}_2(k) = \sum_{n=1}^N d_{12}(n) x_1(k-n) + \sum_{n=1}^N d_{22}(n) x_2(k-n) + \sum_{n=1}^N d_{32}(n) x_3(k-n)$$

$$\hat{x}_3(k) = \sum_{n=1}^N d_{13}(n) x_1(k-n) + \sum_{n=1}^N d_{23}(n) x_2(k-n) + \sum_{n=1}^N d_{33}(n) x_3(k-n)$$

Equation (2-55) can also be written in matrix form.

$$\hat{\underline{x}}(k)^T = \underline{D}^T \underline{x}(k) = [\underline{x}(k)^T \underline{D}]^T \quad (2-56)$$

where

$$\hat{\underline{x}}(k)^T = [x_1(k) \ x_2(k) \ \dots \ x_Q(k)] \quad (Q \times 1)$$

$$\underline{x}^T(k) = [\underline{x}_1^T(k) \ \underline{x}_2^T(k) \ \dots \ \underline{x}_Q^T(k)] \quad (NQ \times 1)$$

$$\underline{x}_i^T(k) = [x_i(k-1) \ x_i(k-2) \ \dots \ x_i(k-N)] \quad (N \times 1)$$

$$\mathbf{D} = \begin{bmatrix} d_{11} & d_{12} & \dots & d_{1N} \\ d_{21} & d_{22} & \dots & d_{2N} \\ \vdots & & & \vdots \\ d_{Q1} & d_{Q2} & \dots & d_{QN} \end{bmatrix} \quad (N \times NQ)$$

$$\underline{d}_{ij}^T = [d_{ij}(1) \ d_{ij}(2) \ \dots \ d_{ij}(N)] \quad (N \times 1)$$

The error of such an estimate is the vector difference in the actual output vector and the predicted output vector.

$$\underline{e}(k) = \underline{x}(k) - \hat{\underline{x}}(k) = [\mathbf{I} - \mathbf{D}^T] \underline{\mathbf{x}}(k)$$

The prediction error covariance matrix is a (QxQ) matrix

$$\underline{P} = \epsilon[\underline{e}(k) \ \underline{e}^T(k)] \quad (2-57)$$

If the trace of the prediction error covariance matrix is minimized the result is

$$\mathbf{R} \mathbf{D} = \mathbf{r} \quad (2-58)$$

where

$$\mathbf{R} = \varepsilon \{ \underline{\mathbf{x}}(k) \underline{\mathbf{x}}^T(k) \} = \varepsilon \begin{bmatrix} \underline{x}_1(k) \\ \vdots \\ \underline{x}_Q(k) \end{bmatrix} [\underline{x}_1(k) \dots \underline{x}_Q(k)]^T$$

$$= \begin{bmatrix} R_{\underline{x}_1 \underline{x}_1}^T & \dots & R_{\underline{x}_1 \underline{x}_Q}^T \\ \vdots & & \vdots \\ R_{\underline{x}_Q \underline{x}_1}^T & \dots & R_{\underline{x}_Q \underline{x}_Q}^T \end{bmatrix} \quad (2-59)$$

is a matrix of autocorrelation and cross correlation matrices and

$$\mathbf{r} = \varepsilon \begin{bmatrix} \underline{x}_1(k) \\ \vdots \\ \underline{x}_Q(k) \end{bmatrix} [\underline{x}_1(k) \dots \underline{x}_Q(k)] = \begin{bmatrix} r_{\underline{x}_1 \underline{x}_1} & \dots & r_{\underline{x}_1 \underline{x}_Q} \\ \vdots & & \vdots \\ r_{\underline{x}_Q \underline{x}_1} & \dots & r_{\underline{x}_Q \underline{x}_Q} \end{bmatrix} \quad (2-60)$$

is a cross correlation matrix of size $(NQ \times N)$ and elements

$$r_{\underline{x}_i \underline{x}_j} = \varepsilon \begin{bmatrix} \underline{x}_i(k-1) \\ \vdots \\ \underline{x}_i(k-N) \end{bmatrix} \underline{x}_j(k) = \begin{bmatrix} R_{\underline{x}_i \underline{x}_j}(1) \\ \vdots \\ R_{\underline{x}_i \underline{x}_j}(N) \end{bmatrix} \quad (2-61)$$

As imposing as this may seem, it is no more than a vector generalization of (2-12) from the AR problem and (2-40) from the MA problem. Each correlation coefficient is replaced with a correlation matrix in the multichannel generalization.

It will be useful in the sequel if the two channel lattice is examined additionally. Equation (2-56) expanded for two channels is

$$\begin{bmatrix} \hat{x}_1(k) \\ \hat{x}_2(k) \end{bmatrix} = \mathbf{D}^T \mathbf{x}(k) = \begin{bmatrix} d_{11} & d_{21} \\ d_{12} & d_{22} \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix}$$

$$= \sum_{n=1}^N \begin{bmatrix} d_{11}(n) & d_{21}(n) \\ d_{12}(n) & d_{22}(n) \end{bmatrix} \begin{bmatrix} x_1(k-n) \\ x_2(k-n) \end{bmatrix} \quad (2-62)$$

The two channel error is the vector difference in the signal to be estimated and the estimation.

$$\underline{e}(k) = \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} - \begin{bmatrix} \hat{x}_1(k) \\ \hat{x}_2(k) \end{bmatrix} = \begin{bmatrix} (x_1(k) - \hat{x}_1(k)) \\ (x_2(k) - \hat{x}_2(k)) \end{bmatrix} \quad (2-63)$$

Equation (2-58), the result of a minimization of the error equations, is in two channels.

$$\begin{bmatrix} \underline{R}_{x_1 x_1} & \underline{R}_{x_1 x_2} \\ \underline{R}_{x_1 x_2} & \underline{R}_{x_2 x_2} \end{bmatrix} \begin{bmatrix} \underline{d}_{11} & \underline{d}_{12} \\ \underline{d}_{21} & \underline{d}_{22} \end{bmatrix} = \begin{bmatrix} \underline{r}_{x_1 x_1} & \underline{r}_{x_1 x_2 p} \\ \underline{r}_{x_2 x_1 n} & \underline{r}_{x_2 x_2} \end{bmatrix} \quad (2-64)$$

where

$$\underline{R}_{x_i x_j} = \begin{bmatrix} R_{x_i x_j}^{(0)} & \dots & R_{x_i x_j}^{(1-N)} \\ \vdots & & \vdots \\ R_{x_i x_j}^{(1-N)} & \dots & R_{x_i x_j}^{(0)} \end{bmatrix}$$

$$\underline{r}_{x_1 x_2} = \begin{bmatrix} R_{x_1 x_1}^{(-1)} \\ \vdots \\ R_{x_1 x_1}^{(-N)} \end{bmatrix}$$

$$\underline{r}_{x_1 x_2 p} = \begin{bmatrix} R_{x_1 x_2}^{(1)} \\ \vdots \\ R_{x_1 x_2}^{(N)} \end{bmatrix}$$

$$\underline{r}_{x_1 x_2 n} = \begin{bmatrix} R_{x_1 x_2}^{(-1)} \\ \vdots \\ R_{x_1 x_2}^{(-N)} \end{bmatrix}$$

Important to the solution of (2-58) is the form of the $(NQ \times NQ)$ correlation matrix \mathbf{R} . The matrices on the main diagonal are all correlation matrices of the form $\mathbf{R}_{\underline{x}_i \underline{x}_i}^T$ and so are symmetric and Toeplitz. The off diagonal matrices are cross correlation matrices that are Toeplitz but not symmetric. Symmetrically opposed matrices are, however, transposes of one another, with the result that the matrix \mathbf{R} is also symmetric. This block Toeplitz form is amenable to a block form of Levinson algorithm solution.

The vector generalization of the AR lattice is shown in Figure 2.8 and is described by the equations

$$\underline{e}^{(n+1)}(k) = \underline{e}^{(n)}(k) - \underline{K}^{(n+1)T} \underline{\bar{e}}^{(n)}(k-1) \quad (2-65)$$

$$\underline{\bar{e}}^{(n+1)}(k) = \underline{\bar{e}}^{(n)}(k-1) - \underline{\bar{K}}^{(n+1)T} \underline{e}^{(n)}(k) \quad (2-66)$$

and the initial conditions

$$\underline{e}^{(0)}(k) = \underline{\bar{e}}^{(0)}(k) = \underline{x}(k) \quad (2-67)$$

The signal paths of the single channel AR model have been replaced with parallel vector signal paths, adders with vector adders and multipliers with matrix multipliers.

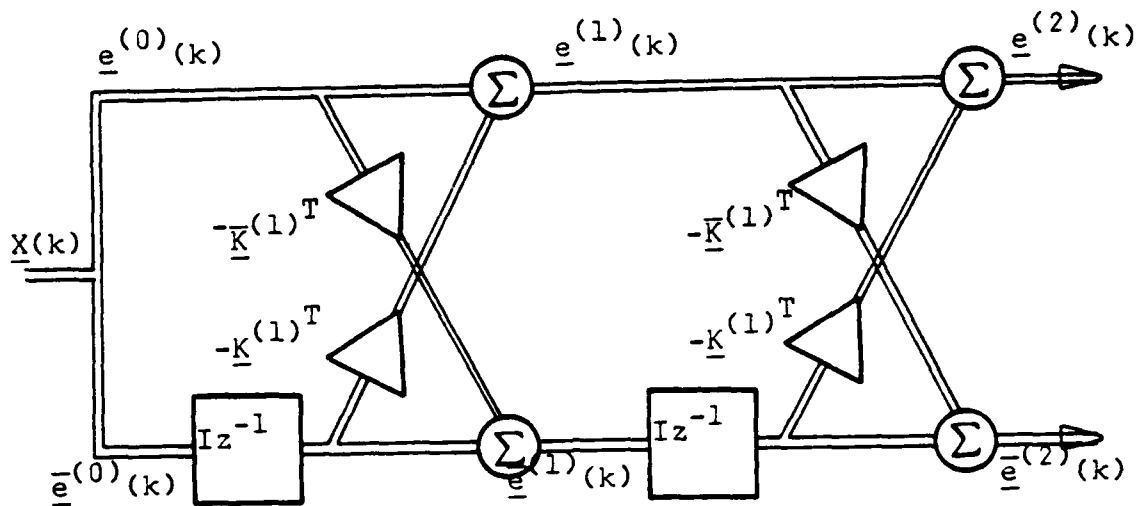


Figure 2.8 A Two Stage Multichannel AR Lattice

Lattice predictor coefficients $\underline{K}^{(n)}$ and $\underline{\bar{K}}^{(n)}$ may be also found from vector generalizations of the single channel case using

$$\underline{K}^{(n+1)} = \underline{P}^{(n)-1} \underline{\Delta}^{(n)T} \quad (2-68)$$

$$\underline{\bar{K}}^{(n+1)} = \underline{P}^{(n)-1} \underline{\Delta}^{(n)} \quad (2-69)$$

$$\underline{\Delta}^{(n)} = \epsilon \{ \underline{e}^{(n)}(k) \underline{\bar{e}}^{(n)T}(k-1) \} \quad (2-70)$$

$$\underline{P}^{(n)} = \epsilon \{ \underline{e}^{(n)}(k) \underline{e}^{(n)T}(k) \} = \underline{P}^{(n-1)} [I - \underline{\bar{K}}^{(n)} \underline{K}^{(n)}] \quad (2-71)$$

$$\underline{\bar{P}}^{(n)} = \epsilon \{ \underline{\bar{e}}^{(n)}(k) \underline{\bar{e}}^{(n)T}(k) \} = \underline{\bar{P}}^{(n-1)} [I - \underline{K}^{(n)} \underline{\bar{K}}^{(n)}] \quad (2-72)$$

As in the single channel case, the optimum predictor for all orders less than N are found with the N -th order solution. Also, orthogonality applies to the backward channel error signals, or

$$\varepsilon \{ \underline{\bar{e}}^i(k) \underline{\bar{e}}^j{}^T(k) \} = \begin{cases} 0 & i \neq j \\ \underline{\bar{P}}^{(i)} & i = j \end{cases}$$

and successive stages are thereby decoupled.

For a multichannel MA model equations (2-65) to (2-72) are used with a vector generalization of (2-54).

$$\hat{\underline{y}}^{(n+1)}(k) = \hat{\underline{y}}^{(n)}(k) + \underline{G}^{(n+1)T} \underline{\bar{e}}^{(n+1)}(k) \quad (2-73)$$

The error of the MA model is expressed by

$$\underline{e}_0^{(n)}(k) = \underline{y}(k) - \hat{\underline{y}}(k)$$

and can be minimized with the result

$$\underline{G}^{(n)} = \underline{\bar{P}}^{(n)-1} \varepsilon \{ \underline{\bar{e}}^{(n)}(k) \underline{y}^T(k) \} \quad (2-74)$$

Figure 2.9 illustrates the resultant structure.

In the next chapter the multichannel generalization is applied to ARMA modeling.

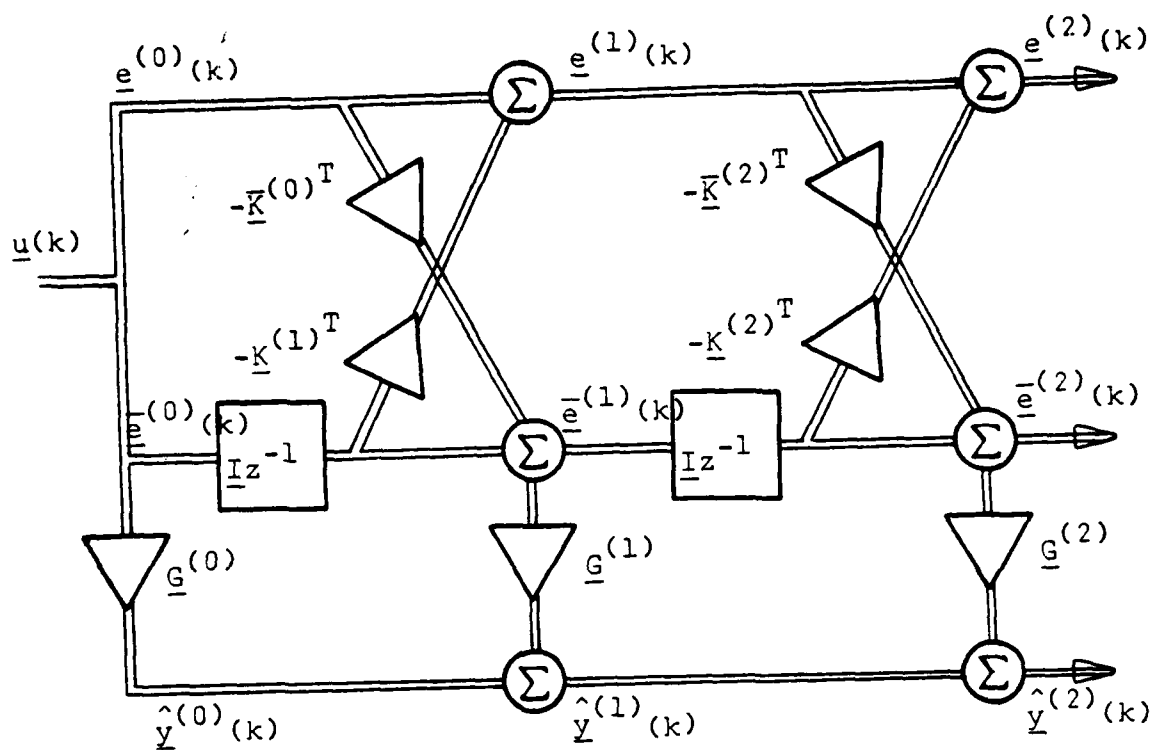


Figure 2.9 A Two Stage Multichannel MA Lattice

III. THE AUTOREGRESSIVE MOVING AVERAGE MODEL IN TIME AND FREQUENCY DOMAIN

While the MA and AR models are suitable for many applications, modeling some systems can require a very high order model to obtain suitable results. Consider the geometric series expansion.

$$\frac{1}{(1 - az^{-1})} = \sum_{n=0}^{\infty} a^n z^{-n} \quad |az^{-1}| < 1$$

This shows that a single pole is equal to an infinite number of zeros, and conversely a single zero is equal to an infinite number of poles. Thus a system with a single pole will require an AR model of infinite order, as would a system with a single zero require an infinite order MA model, to be precisely represented. The ARMA model emulates both poles and zeros, and so can model pole-zero systems without resorting to infinite order solutions.

This chapter reviews contemporary ARMA lattice structures, then develops some new formulations using frequency domain data and considers their implementation as a performance evaluation tool.

A. THE ARMA EQUATIONS

To model a system with both poles and zeros it is intuitively appealing to construct a model that also has both poles and zeros. As was earlier discussed, the equation error model will result in linear equations as the MMSE solution. If the condition that the order of poles equal zeros is imposed the equation error model output estimate is described by

$$\hat{y}(k) = \sum_{n=0}^N a_n u(k-n) + \sum_{n=1}^N b_n y(k-n) \quad (3-1)$$

$$= [\underline{y}^T(k) \mid \underline{u}^T(k)] \begin{bmatrix} \underline{b} \\ \hline \underline{a} \end{bmatrix}$$

where vectors $\underline{y}(k)$, $\underline{u}(k)$, \underline{b} and \underline{a} are defined and dimensioned as in the AR and MA discussions of Sections II.A and II.B and $\hat{y}(k)$ is the ARMA output estimate. This constraint is not limiting since some of the coefficients of the numerator or demoninator can be zero. Minimizing the mean squared error with respect to the parameter vectors \underline{a} and \underline{b} yields

$$\begin{bmatrix} R_{yy}(0) & \dots & R_{yy}(1-N) & | & R_{yu}(1) & R_{yu}(0) & \dots & R_{yu}(1-N) \\ \vdots & & \vdots & & \vdots & & & \vdots \\ R_{yy}(N-1) & \dots & R_{yy}(0) & | & R_{yu}(N) & \dots & & R_{yu}(0) \\ \hline R_{uy}(0) & \dots & R_{uy}(-N) & | & R_{uu}(0) & R_{uu}(-1) & \dots & R_{uu}(-N) \\ R_{uy}(-1) & \dots & R_{uy}(1-N) & | & R_{uu}(1) & R_{uu}(0) & \dots & R_{uu}(1-N) \\ \vdots & & \vdots & & \vdots & \vdots & & \vdots \\ R_{uy}(N+1) & \dots & R_{uy}(0) & | & R_{uu}(N) & R_{uu}(N-1) & \dots & R_{uu}(0) \end{bmatrix} \begin{bmatrix} b_1 \\ \vdots \\ b_N \\ \hline a_0 \\ a_1 \\ \vdots \\ a_N \end{bmatrix} = \begin{bmatrix} R_{yy}(1) \\ \vdots \\ R_{yy}(N) \\ \hline R_{uy}(0) \\ \vdots \\ R_{uy}(N) \end{bmatrix}$$

(3-2)

with a MMSE of

$$E_{\min} = R_{yy}(0) - [\underline{b}_{\text{opt}}^T | \underline{a}_{\text{opt}}^T] \begin{bmatrix} R_{yy}(1) \\ \vdots \\ R_{yy}(N) \\ \hline R_{uy}(0) \\ \vdots \\ R_{uy}(N) \end{bmatrix} \quad (3-3)$$

Equation (3-2) can be readily solved for $\underline{b}_{\text{opt}}$ and $\underline{a}_{\text{opt}}$ by matrix manipulation but again a recursive in order solution can be derived.

B. A RECURSIVE IN ORDER ARMA SOLUTION

The $((2N+1) \times (2N+1))$ correlation matrix of (3-2) is neither Toeplitz nor symmetric, but is nearly so. The offending elements are those associated with the a_0 term.

If it is assumed that a_0 can be found by other means, (3-2) can be restructured so that the correlation matrix is block Toeplitz by subtracting the a_0 terms from both sides of the matrix equations (3-2).

$$\begin{bmatrix} R_{yy}(0) & \dots & R_{yy}(1-N) & | & R_{yu}(0) & \dots & R_{yu}(1-N) \\ \vdots & & \vdots & & \vdots & & \vdots \\ R_{yy}(N-1) & \dots & R_{yy}(0) & | & R_{yu}(N-1) & \dots & R_{yu}(0) \\ \hline R_{uy}(0) & \dots & R_{uy}(1-N) & | & R_{uu}(0) & \dots & R_{uu}(1-N) \\ \vdots & & \vdots & & \vdots & & \vdots \\ R_{uy}(N-1) & \dots & R_{uy}(0) & | & R_{uu}(N-1) & \dots & R_{uu}(0) \end{bmatrix} \begin{bmatrix} b_1 \\ \vdots \\ b_N \\ \hline a_1 \\ \vdots \\ a_N \end{bmatrix} = \begin{bmatrix} R_{yy}(1) \\ \vdots \\ R_{yy}(N) \\ \hline R_{uy}(1) \\ \vdots \\ R_{uy}(N) \end{bmatrix} - a_0 \begin{bmatrix} R_{yu}(1) \\ \vdots \\ R_{yu}(N) \\ \hline R_{uu}(1) \\ \vdots \\ R_{uu}(N) \end{bmatrix}$$

(3-4)

Since a_0 is just the DC gain, and can be easily calculated, this is a reasonable assumption. Equation (3-4) can be rewritten using the notation of (2-64), as was used in the discussion of the two channel AR lattice in Section II.C

$$\begin{bmatrix} \underline{R}_{yy} & \underline{R}_{yu} \\ \underline{R}_{yu}^T & \underline{R}_{uu} \end{bmatrix} \begin{bmatrix} \underline{b} \\ \hline \underline{a} \end{bmatrix} = \begin{bmatrix} \underline{r}_{yy} & \underline{r}_{yu_p} \\ \underline{r}_{yu_n} & \underline{r}_{uu} \end{bmatrix} \begin{bmatrix} 1 \\ \hline -a_0 \end{bmatrix} \quad (3-5)$$

where the vector \underline{a} contains the N elements $a_1 \dots a_N$ of the vector \underline{a} . Here use has been made of the identity $\underline{R}_{uv} = \underline{R}_{vu}^T$. This is compared with (2-64) with $x_1 = y$ and $x_2 = u$

$$\begin{bmatrix} R_{yy} & R_{yu} \\ R_{uy} & R_{uu} \end{bmatrix} \begin{bmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{bmatrix} = \begin{bmatrix} r_{uu} & r_{yu_p} \\ r_{yu_n} & r_{uu} \end{bmatrix}$$

To solve (3-5) using the algorithms for the two channel lattice it is only necessary to make the substitution

$$\begin{bmatrix} \underline{b} \\ -\underline{a} \end{bmatrix} = \begin{bmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{bmatrix} \begin{bmatrix} 1 \\ -a_0 \end{bmatrix} \quad (3-6)$$

after solving for the vectors \underline{d}_{ij} in order to find \underline{b} and \underline{a} . Since a_0 is the DC gain of the system to be modeled it can be found using

$$a_0 = \frac{\varepsilon \{y(k)u(k)\}}{\varepsilon \{u(k)u(k)\}} \quad (3-7)$$

1. An ARMA Lattice Structure

Since the equation error ARMA model can be solved in terms of a two channel AR recursive formulation, it is not difficult to modify the AR two channel lattice to represent the ARMA system.

The ARMA error can be written as the difference between the output $y(k)$ and the estimated output $\hat{y}(k)$ or

$$e_0(k) = y(k) - [\underline{y}^T(k) | \underline{u}^T(k)] \begin{bmatrix} \underline{b} \\ -\underline{a} \end{bmatrix} \quad (3-8)$$

from the error equation model of Figure 1.3 and (3-1). The AR error can be written by substituting (2-62) into (2-63), with $y(k)$ for $x_1(k)$ and $u(k)$ for $x_2(k)$, and transposing, so that

$$\begin{aligned} \underline{e}(k)^T &= \begin{bmatrix} y(k) \\ u(k) \end{bmatrix}^T - [\underline{y}(k)^T | \underline{u}'(k)^T] \begin{bmatrix} \underline{d}_{11} & \underline{d}_{12} \\ \underline{d}_{21} & \underline{d}_{22} \end{bmatrix} \\ &= [e_y(k) \ e_u(k)]^T \end{aligned} \quad (3-9)$$

where $\underline{u}'(k)$ is an N element vector containing the elements $u(k-1) \dots u(k-N)$ of the vector $\underline{u}(k)$.

If (3-9) is post multiplied by a vector $\underline{\psi}$ where

$$\underline{\psi} = \begin{bmatrix} 1 \\ -a_0 \end{bmatrix} \quad (3-10)$$

the result is

$$\begin{aligned} \underline{e}^T(k) \underline{\psi} &= [e_y(k) \ e_u(k)] \begin{bmatrix} 1 \\ -a_0 \end{bmatrix} \\ &= y(k) - a_0 u(k) - [\underline{y}(k) \ \underline{u}'^T(k)] \begin{bmatrix} \underline{b} \\ -\underline{a} \end{bmatrix} = e_0(k) \end{aligned} \quad (3-11)$$

which is exactly the ARMA error of (3-8). Furthermore, the mean squared ARMA error is

$$\begin{aligned}
 \varepsilon\{e_0^2(k)\} &= \underline{\psi}^T \varepsilon\{\underline{e}(k) \underline{e}(k)^T\} \underline{\psi} \\
 &= [1 \ -a_0] \varepsilon \left\{ \begin{bmatrix} e_y(k) & e_y(k) & e_y(k) & e_u(k) \\ e_u(k) & e_y(k) & e_u(k) & e_u(k) \end{bmatrix} \right\} \begin{bmatrix} 1 \\ -a_0 \end{bmatrix} \\
 &= \underline{\psi}^T \underline{P} \underline{\psi} \tag{3-12}
 \end{aligned}$$

where \underline{P} is the forward prediction error covariance matrix of the two channel AR model. If (3-12) is minimized with respect to a_0 the result is

$$a_0 = \frac{\varepsilon\{e_u(k)e_y(k)\}}{\varepsilon\{e_u^2(k)\}} \tag{3-13}$$

which is an alternative solution for a_0 .

To construct an ARMA lattice it is only necessary to derive the ARMA error using (3-11) from the AR lattice.

A second order lattice model is shown as Figure 3.1.

From (2-65), forward prediction in the two channel AR lattice can be written

$$\begin{bmatrix} e_y(k) \\ e_u(k) \end{bmatrix}^{(n+1)} = \begin{bmatrix} e_y(k) \\ e_u(k) \end{bmatrix}^{(n)} - \begin{bmatrix} k_{11} & k_{21} \\ k_{12} & k_{22} \end{bmatrix}^{(n+1)} \begin{bmatrix} \bar{e}_y(k-1) \\ \bar{e}_u(k-1) \end{bmatrix}^{(n)} \tag{3-14}$$

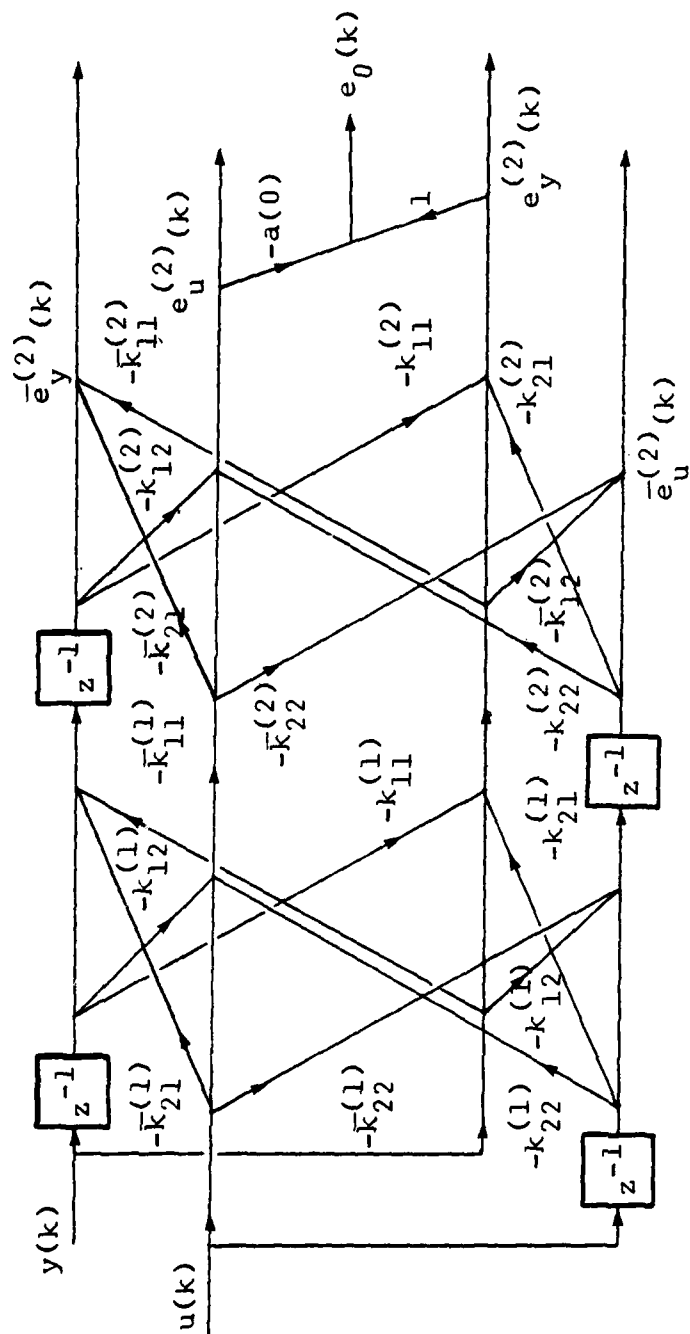


Figure 3.1 A Second Order ARMA Lattice Analysis Structure

From this equation $e_u^{(n)}(k)$ and $e_y^{(n)}(k)$ can be obtained.

$$e_y^{(n)}(k) = e_y^{(n+1)}(k) + k_{11}^{(n+1)} e_y^{(n)}(k-1) + k_{21}^{(n+1)} e_u^{(n)}(k-1) \quad (3-15)$$

$$e_u^{(n)}(k) = e_u^{(n+1)}(k) + k_{12}^{(n+1)} e_y^{(n)}(k-1) + k_{22}^{(n+1)} e_u^{(n)}(k-1) \quad (3-16)$$

Equations (3-15) and (3-16) describe the synthesis structure that can be used to implement the ARMA model as shown by Figure 3.2. For an N-th order realization an input is needed for $e_y^{(N)}(k)$. From (3-11)

$$e_y^{(N)} = e_0(k) + a_0 e_u^{(N)}(k) \quad (3-17)$$

and if the ARMA model is an accurate representation $e_0(k)$ will be small so that (3-17) can be rewritten

$$e_y^{(N)}(k) = a_0 e_u^{(N)}(k) \quad (3-18)$$

This provides the input to $e_y(k)$ shown in Figure 3.2.

C. LATTICE ALGORITHMS IN THE FREQUENCY DOMAIN

In the foregoing it has been tacitly assumed that the data available for analysis is in the form of a finite length time domain signal. This may not be the case. The

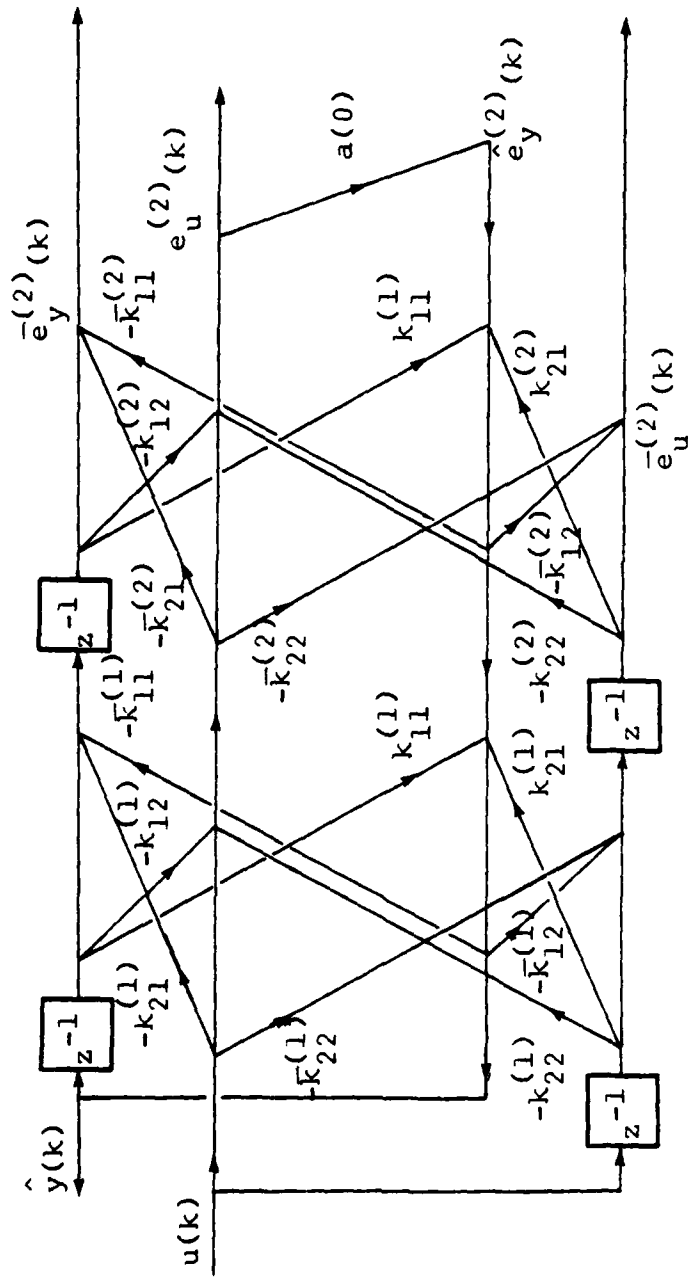


Figure 3.2 A Second Order ARMA Lattice Synthesis Structure

data could be in the form of the Fourier transform of the time domain signal, or it may prove advantageous to process the data in that form. Makhoul [9] has shown that linear prediction can be done equivalently in either time and frequency domain, but there has been in the literature no exploitation of this for the lattice algorithms. Thus it seems useful to examine the lattice structures when the data is represented as a frequency domain spectrum.

1. The Single Channel AR Lattice

The time domain lattice was formulated using the time domain equations (2-24) and (2-25). With frequency domain data available the single channel AR lattice can be formulated using (2-23) and (2-23a). Its structure is shown as Figure 3.3. A unit time delay becomes a multiplication by $e^{-j\omega}$ with the input signal represented by the continuous spectrum $Y(\omega)$. Note that in the backward channel the signal $\bar{E}^{(n)}(\omega)$ is a delayed version of $\bar{E}^{(n)}(\omega)$. The frequency domain error of a zero order prediction is

$$E^{(0)}(\omega) = \bar{E}^{(0)}(\omega) = Y(\omega) \quad (3-19)$$

From the forward error channel recursion (2-23) the magnitude squared error of order $(n+1)$ can be written as

$$E^{(n+1)}(\omega) E^{(n+1)*}(\omega) = [E^{(n)}(\omega) - k^{(n+1)} \bar{E}^{(n)}(\omega)] [E^{(n)*}(\omega) - k^{(n+1)*} \bar{E}^{(n)*}(\omega)] \quad (3-20)$$

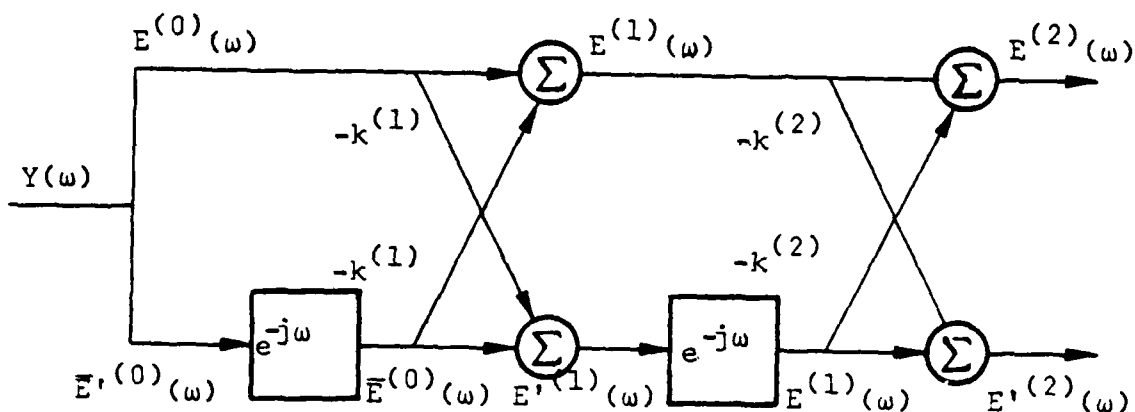


Figure 3.3 A Second Order AR Lattice in the Frequency

where E^* indicates complex conjugate. The total magnitude squared error can be found as the integral of (3-20)

$$\begin{aligned}
 \int_{-\infty}^{\infty} |E^{(n+1)}(\omega)|^2 d\omega &= \int_{-\infty}^{\infty} |E^{(n)}(\omega)|^2 d\omega \\
 &- k^{(n+1)} \left[\int_{-\infty}^{\infty} [E^{(n)}(\omega) \bar{E}^{(n)*}(\omega)] d\omega + \int_{-\infty}^{\infty} [E^{(n)*}(\omega) \bar{E}^{(n)}(\omega)] d\omega \right] \\
 &+ k^{(n+1)^2} \int_{-\infty}^{\infty} |\bar{E}^{(n)}(\omega)|^2 d\omega
 \end{aligned} \tag{3-21}$$

Since $XZ^* = [X^*Z]^*$ the bracketed term of the right side of (3-21) can be rewritten as

$$\begin{aligned}
& \left[\int_{-\infty}^{\infty} [E^{(n)}(\omega) \bar{E}^{(n)*}(\omega)] d\omega + \int_{-\infty}^{\infty} [E^{(n)*}(\omega) \bar{E}^{(n)}(\omega)] d\omega \right] \\
& = \int_{-\infty}^{\infty} [E^{(n)}(\omega) \bar{E}^{(n)*}(\omega)] d\omega + \int_{-\infty}^{\infty} [E^{(n)}(\omega) \bar{E}^{(n)*}(\omega)]^* d\omega
\end{aligned}
\tag{3-22}$$

Since $Y(\omega)$ is the spectrum of a stable and causal sequence then $\text{Re}\{Y(\omega)\}$ is an even function and $\text{Im}\{Y(\omega)\}$ is an odd function. This is also true for the error signals $E^{(n)}(\omega)$ and $\bar{E}^{(n)}(\omega)$. The result of the integrations of (3-22), must therefore be real and the two terms of (3-22) are equal, or

$$\begin{aligned}
& \int_{-\infty}^{\infty} [E^{(n)}(\omega) \bar{E}^{(n)*}(\omega)] d\omega + \int_{-\infty}^{\infty} [E^{(n)*}(\omega) \bar{E}^{(n)}(\omega)] d\omega \\
& = 2 \int_{-\infty}^{\infty} [E^{(n)}(\omega) \bar{E}^{(n)*}(\omega)] d\omega = 2 \int_{-\infty}^{\infty} [E^{(n)*}(\omega) \bar{E}^{(n)}(\omega)] d\omega
\end{aligned}
\tag{3-23}$$

Substituting (3-23) for the middle term of (3-21) and minimizing the result with respect to $k^{(n+1)}$ yields

$$k^{(n+1)} = \frac{\int_{-\infty}^{\infty} [E^{(n)}(\omega) \bar{E}^{(n)*}(\omega)] d\omega}{\int_{-\infty}^{\infty} |\bar{E}^{(n)}(\omega)|^2 d\omega} \quad \text{for a continuous spectrum } Y(\omega) \tag{3-24}$$

or equivalently

$$k^{(n+1)} = \frac{\int_{-\pi}^{\pi} [E^{(n)}(\omega) \bar{E}^{(n)*}(\omega)] d\omega}{\int_{-\pi}^{\pi} |E^{(n)}(\omega)|^2 d\omega} \quad \begin{array}{l} \text{for a periodic} \\ \text{spectrum } Y(\omega) \\ \text{with period } 2\pi \end{array} \quad (3-25)$$

or

$$k^{(n+1)} = \frac{\sum_{k=0}^{m-1} E^{(n)}\left(\frac{k}{mT}\right) \bar{E}^{(n)*}\left(\frac{k}{mT}\right)}{\sum_{k=0}^{m-1} |E^{(n)}\left(\frac{k}{mT}\right)|^2} \quad \begin{array}{l} \text{for a discrete} \\ \text{periodic } m\text{-point} \\ \text{spectrum } Y\left(\frac{k}{mT}\right) \\ \text{with period } mT \end{array} \quad (3-26)$$

If the discrete correlation theorem and Parseval's theorem are applied to (3-26) the result is

$$k^{(n+1)} = \frac{\varepsilon\{e^{(n)}(k) \bar{e}^{(n)}(k-1)\}}{\varepsilon\{\bar{e}^{(n)2}(k)\}} = k_F^{(n+1)} \quad (3-27)$$

which is equivalent to the forward solution (2-27).

The same procedure can be applied to the backward error recursion (2-23a) with the result

$$k^{(n+1)} = \frac{\int_{-\infty}^{\infty} [E^{(n)}(\omega) \bar{E}^{(n)*}(\omega)] d\omega}{\int_{-\infty}^{\infty} |E^{(n)}(\omega)|^2 d\omega} \quad \begin{array}{l} \text{for a continuous} \\ \text{spectrum } Y(\omega) \end{array} \quad (3-28)$$

$$k^{(n+1)} = \frac{\int_{-\pi}^{\pi} [E^{(n)}(\omega) \bar{E}^{(n)*}(\omega)] d\omega}{\int_{-\pi}^{\pi} |E^{(n)}(\omega)|^2 d\omega} \quad \begin{array}{l} \text{for a periodic} \\ \text{spectrum } Y(\omega) \\ \text{with period } 2\pi \end{array} \quad (3-29)$$

$$k^{(n+1)} = \frac{\sum_{k=0}^{m-1} E^{(n)}\left(\frac{k}{mT}\right) \bar{E}^{(n)*}\left(\frac{k}{mT}\right)}{\sum_{k=0}^{m-1} |\bar{E}^{(n)}\left(\frac{k}{mT}\right)|^2} \quad \begin{array}{l} \text{for a discrete} \\ \text{periodic } m\text{-} \\ \text{point spectrum} \\ Y\left(\frac{k}{mT}\right) \text{ with} \\ \text{period } mT \end{array} \quad (3-30)$$

which can be shown to be equivalent to (2-28), the backward equation.

$$k^{(n+1)} = \frac{\epsilon \{e^{(n)}(k) \bar{e}^{(n)}(k-1)\}}{\epsilon \{e^{(n)}(k)\}^2} = k_B^{(n+1)} \quad (3-31)$$

2. The Two Channel ARMA Lattice

The multichannel lattice can be similarly developed using frequency domain transformations of the time domain equations. Doing so, the two channel lattice equations become

$$\underline{E}^{(0)}(\omega) = \underline{\bar{E}}^{(0)}(\omega) = \begin{bmatrix} Y(\omega) \\ \bar{U}(\omega) \end{bmatrix} \quad (3-32)$$

$$\underline{\bar{E}}^{(n)}(\omega) = e^{j\omega} \underline{\bar{E}}^{(n)}(\omega) \quad (3-33)$$

$$\begin{aligned} \underline{\Delta}^{(n)} &= \int \underline{E}^{(n)}(\omega) \underline{\bar{E}}^{(n)*T}(\omega) d\omega \\ &= \begin{bmatrix} \int E_y^{(n)}(\omega) \bar{E}_y^{(n)*}(\omega) d\omega & \int E_y^{(n)}(\omega) \bar{E}_u^{(n)*}(\omega) d\omega \\ \int E_u^{(n)}(\omega) \bar{E}_y^{(n)*}(\omega) d\omega & \int E_u^{(n)}(\omega) \bar{E}_u^{(n)*}(\omega) d\omega \end{bmatrix} \end{aligned} \quad (3-34)$$

$$\begin{aligned} \underline{P}^{(n)} &= \int \underline{E}^{(n)}(\omega) \underline{E}^{(n)*T}(\omega) d\omega \\ &= \begin{bmatrix} \int E_y^{(n)}(\omega) E_y^{(n)*}(\omega) d\omega & \int E_y^{(n)}(\omega) E_u^{(n)*}(\omega) d\omega \\ \int E_u^{(n)}(\omega) E_y^{(n)*}(\omega) d\omega & \int E_u^{(n)}(\omega) E_u^{(n)*}(\omega) d\omega \end{bmatrix} \end{aligned} \quad (3-35)$$

$$\begin{aligned} \underline{\bar{P}}^{(n)} &= \int \underline{\bar{E}}^{(n)}(\omega) \underline{\bar{E}}^{(n)*T}(\omega) d\omega \\ &= \begin{bmatrix} \int \bar{E}_y^{(n)}(\omega) \bar{E}_y^{(n)*}(\omega) d\omega & \int \bar{E}_y^{(n)}(\omega) \bar{E}_u^{(n)*}(\omega) d\omega \\ \int \bar{E}_u^{(n)}(\omega) \bar{E}_y^{(n)*}(\omega) d\omega & \int \bar{E}_u^{(n)}(\omega) \bar{E}_u^{(n)*}(\omega) d\omega \end{bmatrix} \end{aligned} \quad (3-36)$$

$$\underline{K}^{(n+1)} = \underline{P}^{(n)-1} \underline{\Delta}^{(n)T} \quad (3-37)$$

$$\underline{K}^{(n+1)} = \underline{P}^{(n)-1} \underline{\Delta}^{(n)} \quad (3-38)$$

$$\underline{E}^{(n+1)}(\omega) = \underline{E}^{(n)}(\omega) - \underline{K}^{(n+1)T} \underline{\bar{E}}^{(n)}(\omega) \quad (3-39)$$

$$\underline{\bar{E}}^{(n+1)}(\omega) = \underline{\bar{E}}^{(n)}(\omega) - \underline{K}^{(n+1)T} \underline{E}^{(n)}(\omega) \quad (3-40)$$

where the limits of integration depend upon the periodicity of the available data. If the data is in a discrete form the integrations are replaced with summations.

It is important to note that when the data is in a discrete form the result of a correlation obtained for a time shift other than zero will differ when calculated in the frequency domain from the result normally obtained from time domain data. Consider first two data signals $y(k)$ and $x(k)$, both defined for $1 \leq k \leq N$, and k an integer. The correlation of delay ℓ is obtained by the summation

$$R_{yx}(\ell) = \left[\sum_{k=1}^{N-\ell} y(k) x(k+\ell) \right] \frac{1}{N-\ell}$$

and has the effect of windowing the data so that data points outside the window are ignored. If these data signals are transformed to the discrete frequency domain with a digital Fourier transform the resultant spectrum is periodic, which implies that in the time domain the data signal has also been made to be periodic. A correlation obtained in the frequency domain is now

$$R_{yx}(\ell) = \frac{1}{2\pi} \int_{-\pi}^{\pi} Y(\omega) X(-\omega) e^{\frac{j\omega\ell}{N}} d\omega$$

which when transformed back to the time domain is equivalent to

$$R_{yx}(\ell) = \left[\sum_{k=1}^{N-\ell} y(k)x(k+\ell) + \sum_{k=N-\ell+1}^N y(k)x(k+\ell-N) \right] \frac{1}{N-\ell}$$

This is because the effect in the time domain of the frequency domain time delay $e^{\frac{j\omega\ell}{N}}$ is a circular shift of the data in the time domain. As was discussed in the preceding chapter time domain data would normally use a linear shift and then be either windowed or would be considered of infinite length. Either method introduces it's own distortion into the estimation of the correlation. The circular shift of the frequency domain correlation is a third and different kind of distortion. This is important when the number of data points is small, because a bias will be introduced into the estimate of the correlation.

D. SYSTEM IDENTIFICATION WITH THE ARMA LATTICE

The problem of system identification concerns the determination of a mathematical model of a system from its input output relationship. An impetus for this research has been the issue of fault detection within a system. If a model can be constructed of an operating system, either on line or under test, the model parameters might reveal if the system is functioning normally and if not, what class of component is faulty. While the ARMA lattice has characteristics which make it particularly attractive for this application, particularly its robustness, there are some characteristics that still deserve attention.

In system identification the lattice model would be applied as shown in Figure 3.4. The lattice parameters $\underline{K}^{(n)}$, $\underline{\bar{K}}^{(n)}$ and a_0 that result from a test can be compared with a dictionary of parameters obtained under normal and faulty conditions. Using lattice parameters directly, though, results in more model parameters than if the vectors \underline{a} and \underline{b} were used for identification. Identification using \underline{a} and \underline{b} uses $(2n+1)$ parameters for an n -th order system while lattice parameters result in $(8n+1)$ parameters. Therefore it is beneficial to reduce the number of lattice parameters that must be examined in the identification problem.

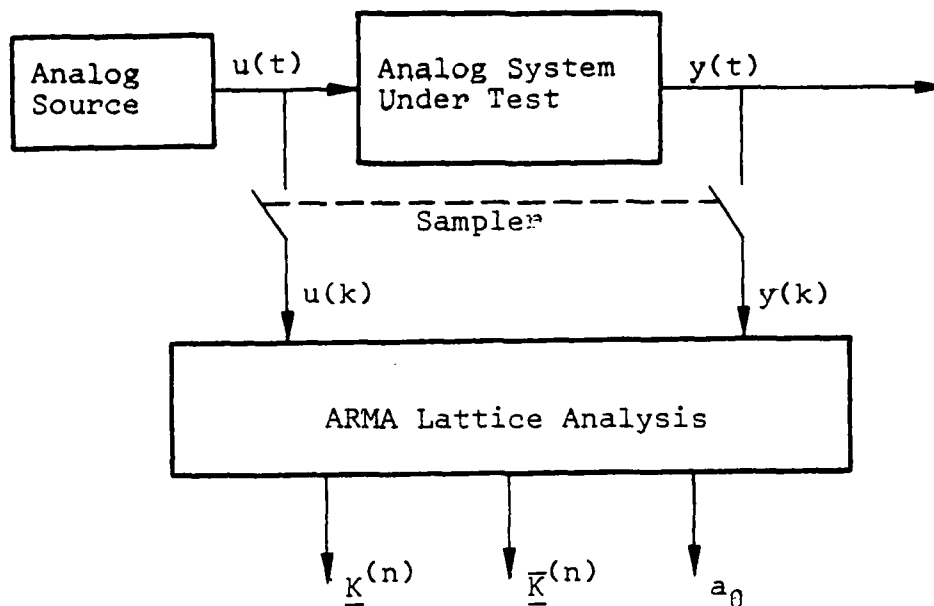


Figure 3.4 ARMA Lattice Used for System Identification

1. Restraints on ARMA Lattice Parameters

ARMA lattice parameters are a function of input signal second order statistics. The input signals $u(k)$ and $y(k)$ to the ARMA lattice are related by the transfer function of the system under test. This suggests that the ARMA lattice parameters are interrelated. To examine the problem the equations for the lattice parameters will be

expanded in terms of the system input $u(k)$ and output $y(k)$.
The vector defining the initial conditions for the ARMA
lattice first stage is

$$\underline{X}(k) = \begin{bmatrix} y(k) \\ u(k) \end{bmatrix} = \underline{e}^{(0)}(k) = \underline{\bar{e}}^{(0)}(k) \quad (3-41)$$

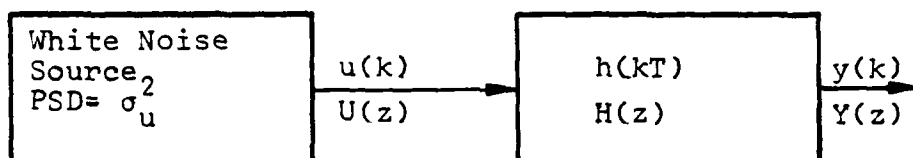
The forward and backward prediction error covariance
matrix become

$$\begin{aligned} \underline{P}^{(0)} &= \underline{\bar{P}}^{(0)} = \varepsilon[\underline{e}^{(0)}(k)\underline{e}^{(0)T}(k)] \\ &= \varepsilon \begin{bmatrix} y^2(k) & y(k)u(k) \\ y(k)u(k) & u^2(k) \end{bmatrix} \end{aligned} \quad (3-42)$$

and the matrix $\underline{\Delta}^{(0)}$ becomes

$$\begin{aligned} \underline{\Delta}^{(0)} &= \varepsilon[\underline{e}^{(0)}(k)\underline{\bar{e}}^{(0)T}(k-1)] \\ &= \varepsilon \begin{bmatrix} y(k)y(k-1) & y(k)u(k-1) \\ u(k)y(k-1) & u(k)u(k-1) \end{bmatrix} \end{aligned} \quad (3-43)$$

Now consider the result when the signals $u(k)$ and $y(k)$ are from a system as shown in Figure 3.5



$$H(z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2} + \dots}{1 + b_1 z^{-1} + b_2 z^{-2} + \dots}$$

Figure 3.5 System Excited with White Noise

From linear system analysis theory several statements can be made about the second order statistics of $y(k)$ and $u(k)$, assuming that $u(k)$ is white noise.

$$E\{u(k)u(k+l)\} = R_{uu}(l) = \begin{cases} \sigma_u^2 & \text{for } l=0 \\ 0 & \text{for } l \neq 0 \end{cases} \quad (3-44)$$

$$\begin{aligned} E\{y(k)y(k+l)\} &= R_{yy}(l) = R_{yy}(-l) \\ &= \sigma_u^2 \sum_{k=0}^{\infty} h(k)h(k+l) \end{aligned} \quad (3-45)$$

$$E\{u(k)y(k+l)\} = R_{uy}(l) = \begin{cases} \sigma_u^2 h(l) & l \geq 0 \\ 0 & l < 0 \end{cases} \quad (3-46)$$

$$E\{y(k)u(k+l)\} = R_{yu}(l) = R_{uy}(-l) = \begin{cases} \sigma_u^2 h(-l) & l \leq 0 \\ 0 & l > 0 \end{cases} \quad (3-47)$$

$$h(0) = a_0 \quad (3-48)$$

These identities can be substituted into (3-42) and (3-43). Thus

$$\underline{P}^{(0)} = \underline{\bar{P}}^{(0)} = \begin{bmatrix} R_{yy}(0) & R_{yu}(0) \\ R_{yu}(0) & R_{uu}(0) \end{bmatrix} = \begin{bmatrix} R_{yy}(0) & \sigma_u^2 a_0 \\ \sigma_u^2 a_0 & \sigma_u^2 \end{bmatrix} \quad (3-49)$$

$$\underline{\Delta}^{(0)} = \begin{bmatrix} R_{yy}(1) & R_{yu}(-1) \\ R_{yu}(-1) & R_{uu}(-1) \end{bmatrix} = \begin{bmatrix} R_{yy}(1) & \sigma_u^2 h(1) \\ 0 & 0 \end{bmatrix} \quad (3-50)$$

With these, the lattice parameters can be found

$$\underline{k}^{(1)} = \underline{P}^{(0)-1} \underline{\Delta}^{(0)T} = \frac{1}{\text{Det}(\underline{P}^{(0)})} \begin{bmatrix} \sigma_u^2 & -\sigma_u^2 a_0 \\ -\sigma_u^2 a_0 & R_{yy}(0) \end{bmatrix} \begin{bmatrix} R_{yy}(1) & 0 \\ \sigma_u^2 h(1) & 0 \end{bmatrix}$$

$$= \frac{1}{\text{Det}(\underline{P}^{(0)})} \begin{bmatrix} \sigma_u^2 (R_{yy}(1) - a_0 \sigma_u^2 h(1)) & 0 \\ \sigma_u^2 (R_{yy}(0) h(1) - a_0 R_{yy}(1)) & 0 \end{bmatrix} \quad (3-51)$$

$$\underline{\bar{k}}^{(1)} = \underline{P}^{(0)-1} \underline{\Delta}^{(0)} = \frac{1}{\text{Det}(\underline{P}^{(0)})} \begin{bmatrix} \sigma_u & -\sigma_u^2 a_0 \\ -\sigma_u^2 a_0 & R_{yy}(0) \end{bmatrix} \begin{bmatrix} R_{yy}(1) & \sigma_u^2 h(1) \\ 0 & 0 \end{bmatrix}$$

$$= \frac{1}{\text{Det}(\underline{P}^{(0)})} \begin{bmatrix} \sigma_u^2 R_{yy}(1) & \sigma_u^4 h(1) \\ -\sigma_u^2 a_0 R_{yy}(1) & -\sigma_u^4 a_0 h(1) \end{bmatrix} \quad (3-52)$$

From (3-51) and (3-52) some restraints on the lattice parameters are evident.

$$k_{12}^{(1)} = k_{22}^{(1)} = 0 \quad (3-53)$$

$$\bar{k}_{21}^{(1)} = -a_0 \bar{k}_{11}^{(1)} \quad (3-54)$$

$$\bar{k}_{22}^{(1)} = a_0 \bar{k}_{12}^{(1)} \quad (3-55)$$

$$k_{11}^{(1)} = \bar{k}_{11}^{(1)} + \bar{k}_{22}^{(1)} \quad (3-56)$$

It can thus be seen that it is not necessary to calculate all eight lattice parameters. Only three, plus the DC gain a_0 , are needed. Alternatively, the parameter matrices can be calculated and either (3-54) or (3-55) used for an alternative equation for calculation of a_0 .

Equation (3-53) can be interpreted in light of the lattice structure. $k_{12}^{(1)}$ and $k_{22}^{(1)}$ are the coefficients that are the predictors of the future value of $u(k)$. Since $u(k)$ is uncorrelated from sample to sample no prediction can be made and these must be zero.

Some of these relations can be extended to later stages. The forward error vector of the first stage is

$$\underline{e}^{(1)}(k) = \begin{bmatrix} \bar{F}_1 \{y(k-1), y(k), u(k)\} \\ u(k) \end{bmatrix} \quad (3-57)$$

because of (3-53) while the backward error vector is

$$\underline{\bar{e}}^{(1)}(k-1) = \begin{bmatrix} F_2\{y(k-1), y(k-2), u(k-2)\} \\ F_3\{y(k-2), u(k-1), y(k-1)\} \end{bmatrix} \quad (3-58)$$

where $F_i\{\cdot\}$ indicates a linear function of the arguments.

$\underline{\Delta}^{(1)}$ is now defined as

$$\begin{aligned} \underline{\Delta}^{(1)} &= \begin{bmatrix} \Delta_{11}^{(1)} \\ \varepsilon\{u(k) \cdot F_2\{y(k-1), y(k-2), u(k-2)\}\} \\ \Delta_{12}^{(1)} \\ \varepsilon\{u(k) \cdot F_3\{y(k-2), u(k-1), y(k-1)\}\} \end{bmatrix} \\ &= \begin{bmatrix} \Delta_{11}^{(1)} & \Delta_{12}^{(1)} \\ 0 & 0 \end{bmatrix} \end{aligned} \quad (3-59)$$

because of the independence of samples of $u(k)$. When this is applied to (2-65) for $\underline{k}^{(2)}$ the result is

$$\begin{aligned} \underline{k}^{(2)} &= \underline{\bar{P}}^{(1)-1} \underline{\Delta}^{(1)T} = \frac{1}{\text{Det}(\underline{\bar{P}}^{(1)})} \begin{bmatrix} \bar{p}_{22}^{(1)} & -\bar{p}_{21}^{(1)} \\ -\bar{p}_{12}^{(1)} & \bar{p}_{11}^{(1)} \end{bmatrix} \begin{bmatrix} \Delta_{11}^{(1)} & 0 \\ \Delta_{12}^{(1)} & 0 \end{bmatrix} \\ &= \begin{bmatrix} k_{11}^{(2)} & 0 \\ k_{12}^{(2)} & 0 \end{bmatrix} \end{aligned} \quad (3-60)$$

By induction this argument can be extended to arbitrary order with the result

$$k_{12}^{(n)} = k_{22}^{(n)} = 0, \quad n = 0, 1, \dots \quad (3-61)$$

when

$$R_{uu}(\ell) = \sigma_u^2 \delta(\ell)$$

Using (3-61) the forward error vector can be written for order n

$$\underline{e}^{(n)}(k) = \begin{bmatrix} a_0 u(k) + F_u \{y(k-1), \dots, y(k-n), u(k-1), \dots, u(k-n)\} \\ u(k) \end{bmatrix} \quad (3-62)$$

and this can be used to find the forward error covariance matrix

$$\begin{aligned} \underline{P}^{(n)} &= E \{ \underline{e}^{(n)}(k) \underline{e}^{(n)T}(k) \} \\ &= \begin{bmatrix} \rho_{11}^{(n)} & E \{ u(k) a_0 u(k) + u(k) \cdot F_u \{y(k-1), \dots, y(k-n), u(k-1), \dots, u(k-n)\} \} \\ \rho_{21}^{(n)} & E \{ u^2(k) \} \end{bmatrix} \\ &= \begin{bmatrix} \rho_{11}^{(n)} & a_0 \sigma_u^2 \\ \rho_{21}^{(n)} & \sigma_u^2 \end{bmatrix} \end{aligned} \quad (3-64)$$

If this is now used with (3-59) of order n to find the backward parameter matrix the result is

$$\begin{aligned}\underline{\bar{K}}^{(n+1)} &= \underline{P}^{(n)-1} \underline{\Delta}^{(n)} = \frac{1}{\text{Det}(\underline{P}^{(n)})} \begin{bmatrix} \sigma_u^2 & -\rho_{21}^{(n)} \\ -a_0 \sigma_u^2 & \rho_{11}^{(n)} \end{bmatrix} \begin{bmatrix} \Delta_{11}^{(n)} & \Delta_{12}^{(n)} \\ 0 & 0 \end{bmatrix} \\ &= \frac{1}{\text{Det}(\underline{P}^{(n)})} \begin{bmatrix} \sigma_u^2 \Delta_{11}^{(n)} & \sigma_u^2 \Delta_{12}^{(n)} \\ -a_0 \sigma_u^2 \Delta_{11}^{(n)} & -a_0 \sigma_u^2 \Delta_{12}^{(n)} \end{bmatrix} \quad (3-64)\end{aligned}$$

and from this it is seen that

$$\bar{k}_{21}^{(n)} = -a_0 \bar{k}_{11}^{(n)} \quad (3-65)$$

$$\bar{k}_{22}^{(n)} = -a_0 \bar{k}_{12}^{(n)} \quad (3-66)$$

For the single channel AR lattice the forward and backward coefficients were equal, but for the multichannel lattice it is known that the forward and backward parameter matrices are not equal. Nuttal [7] has shown that the determinants of the forward and backward error covariance matrices are equal, and from this it can be shown that the determinants of forward and backward lattice parameter matrices are equal.

It is known that $\underline{P}^{(0)} = \underline{\bar{P}}^{(0)}$ so it follows that

$$\text{Det}(\underline{P}^{(0)}) = \text{Det}(\underline{\bar{P}}^{(0)}) \quad (3-67)$$

Also

$$\begin{aligned} \underline{P}^{(n+1)} &= \underline{P}^{(n)} [\underline{I} - \underline{\bar{K}}^{(n+1)} \underline{K}^{(n+1)}] \\ &= \underline{P}^{(n)} \underline{\bar{K}}^{(n+1)} [\underline{\bar{K}}^{(n+1)^{-1}} - \underline{K}^{(n+1)}] \end{aligned} \quad (3-68)$$

and

$$\begin{aligned} \underline{\bar{P}}^{(n+1)} &= \underline{\bar{P}}^{(n)} [\underline{I} - \underline{K}^{(n+1)} \underline{\bar{K}}^{(n+1)}] \\ &= \underline{\bar{P}}^{(n)} [\underline{\bar{K}}^{(n+1)^{-1}} - \underline{K}^{(n+1)}] \underline{\bar{K}}^{(n+1)} \end{aligned} \quad (3-69)$$

It is known that

$$\text{Det}(\underline{AB}) = \text{Det}(\underline{A})\text{Det}(\underline{B}) = \text{Det}(\underline{B})\text{Det}(\underline{A})$$

so that the determinants of (3-68) and (3-69) can be written

$$\text{Det}(\underline{P}^{(n+1)}) = \text{Det}(\underline{P}^{(n)})\text{Det}(\underline{\bar{K}}^{(n+1)})\text{Det}(\underline{\bar{K}}^{(n+1)^{-1}} - \underline{K}^{(n+1)})$$

$$\text{Det}(\underline{\bar{P}}^{(n+1)}) = \text{Det}(\underline{\bar{P}}^{(n)})\text{Det}(\underline{\bar{K}}^{(n+1)})\text{Det}(\underline{\bar{K}}^{(n+1)^{-1}} - \underline{K}^{(n+1)})$$

which shows that $\text{Det}(\underline{P}^{(n+1)}) = \text{Det}(\underline{\bar{P}}^{(n+1)})$ if $\text{Det}(\underline{P}^{(n)}) = \text{Det}(\underline{\bar{P}}^{(n)})$.

In view of (3-67) it can be inductively reasoned that

$$\text{Det}(\underline{P}^{(n)}) = \text{Det}(\underline{\bar{P}}^{(n)}) \quad (3-70)$$

Now the determinants of the lattice parameters defined by (2-68) and (2-69) can be written

$$\text{Det}(\underline{K}^{(n+1)}) = \text{Det}(\underline{\bar{P}}^{(n)-1}) \text{Det}(\underline{\Delta}^{(n)T})$$

$$\text{Det}(\underline{\bar{K}}^{(n+1)}) = \text{Det}(\underline{P}^{(n)-1}) \text{Det}(\underline{\Delta}^{(n)})$$

and since $\text{Det}(\underline{A}) = \text{Det}(\underline{A}^T)$

$$\text{Det}(\underline{K}^{(n+1)}) = \text{Det}(\underline{\bar{K}}^{(n+1)}) \quad (3-71)$$

A summary of these restraints is presented as Table I.

TABLE I
SUMMARY OF RESTRAINTS ON ARMA LATTICE PARAMETERS

<u>Restraint</u>	<u>Conditions</u>
$\text{Det}(\underline{K}^{(n)}) = \text{Det}(\underline{\bar{K}}^{(n)})$	none
$k_{12}^{(n)} = k_{22}^{(n)} = 0$	$R_{uu}(\ell) = \sigma_u \delta(\ell)$
$\bar{k}_{21}^{(n)} = -a_0 \bar{k}_{11}^{(n)}$	$R_{uu}(\ell) = \sigma_u \delta(\ell)$
$\bar{k}_{22}^{(n)} = -a_0 \bar{k}_{12}^{(n)}$	$R_{uu}(\ell) = \sigma_u \delta(\ell)$
$k_{11}^{(1)} = \bar{k}_{11}^{(1)} + \bar{k}_{22}^{(1)}$	$R_{uu}(\ell) = \sigma_u \delta(\ell)$

2. Uniqueness of the Lattice Parameters

A particularly troublesome consideration in using ARMA lattice parameters for system identification is the lack of uniqueness of the lattice parameters for the system under test. This is evident because the lattice parameters are not just functions of the system under examination, but also of the test signal used to drive the system. Thus for different test signals such as an ensemble of finite white noise input signals, an ensemble of lattice parameters are obtained. Interestingly, experiments have shown that the transfer function coefficients calculated using these lattice parameters will generally have less dispersion than the lattice parameters from which they are calculated.

Thus, the selection of a test signal influences both the accuracy of the model and the expectation of the lattice parameters that result. The mean square value of equation error of the ARMA model can be expressed in integral form as

$$E = \frac{1}{2\pi} \int_{-\pi}^{\pi} E(e^{j\omega T}) E^*(e^{j\omega T}) d\omega T \quad (3-72a)$$

The equation error can be written in the z domain as

$$E(z) = [B(z)H(z) - A(z)]U(z) \quad (3-72b)$$

Substituting (3-72b) into (3-72a) results in

$$E = \frac{1}{2\pi} \int_{-\pi}^{\pi} | [B(e^{j\omega T})H(e^{j\omega T}) - A(e^{j\omega T})] |^2 |U(e^{j\omega T})|^2 d\omega T \quad (3-73)$$

Equation (3-73) shows that the power spectrum of the driving source is a weighting function for the resultant modeling error. To obtain a model that is equally accurate throughout the spectrum, a test source must be used which has equal energy throughout the spectrum. An obvious source with a uniform power spectrum is a white noise source. Besides giving equal weighting to the error spectrum, white noise involves, as has been shown in the preceeding section, simplified calculations and a simplified lattice structure. This, however, is not the total answer. When applied to system identification, the correlations and cross correlations of $y(k)$ and $u(k)$ generally will not be available. Instead these quantities will be estimated from finite sample functions of $y(k)$ and $u(k)$. Even though these signals may be ergodic, the values obtained are only estimates of the correlations.

Experiments with a white noise source have shown that different ensembles of input signals will result in different lattice parameters for each member of the ensemble.

In order to improve the ensemble averages obtained for the lattice parameters, some form of input preprocessing can be performed to whiten the input or equivalently flatten it's spectrum.

Consider first Figure 3.6a, which is a modification of the system of Figure 3.4. An adjustable filter with transfer function $H(\omega)$ has been added to the input of the system under test. The transfer function $H(\omega)$ can be adjusted to made $U'(\omega)$ white if the spectrum of the noise source is not white and has not zeros on the ω axis. If the system under test is linear, then the lattice filter derived from Figure 3.6b is equivalent to the one derived from Figure 3.6c. Figure 3.7 is a block diagram for calculating the frequency domain ARMA lattice model with four preprocessing techniques.

- a) Preprocessing in the time domain on both the input and output signals of the system under test.
- b) Preprocessing in the frequency domain on both the input and output signals of the system under test.

Since time frequency domain operations are equivalent they can be interchanged. For comparison they are considered separately to determine if there are any computational advantages of one over the other. It should be noted that as long as equivalent operations are performed on both input and output, as discussed earlier, the resultant lattice model will still model the system under test, though the specific lattice coefficients may differ.

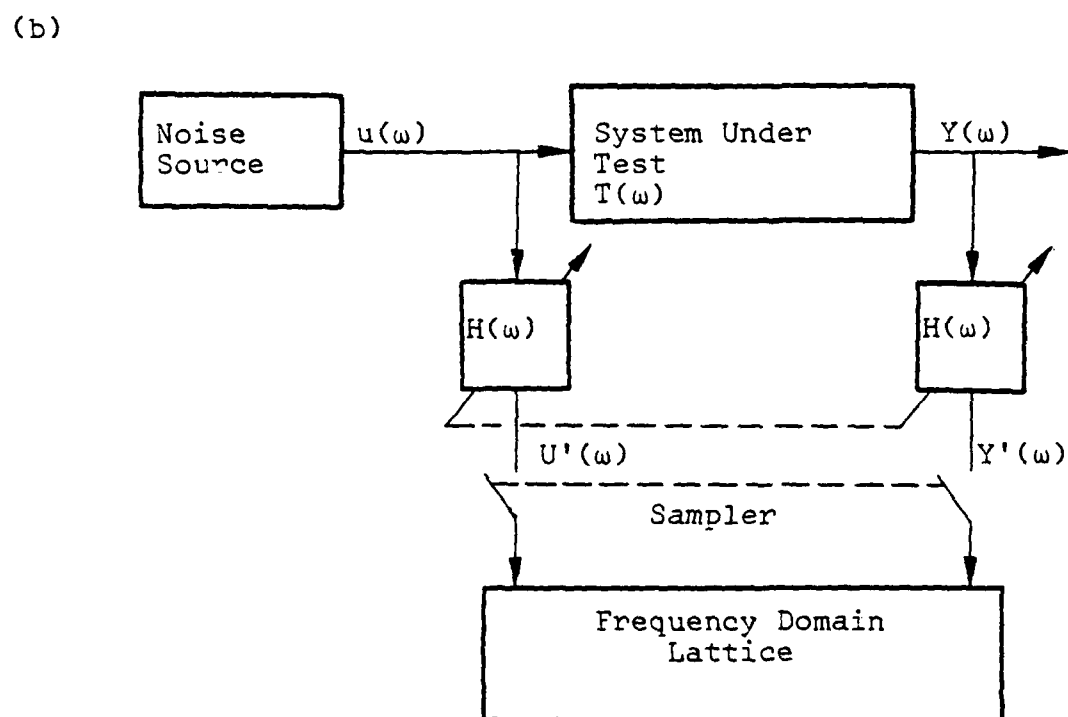
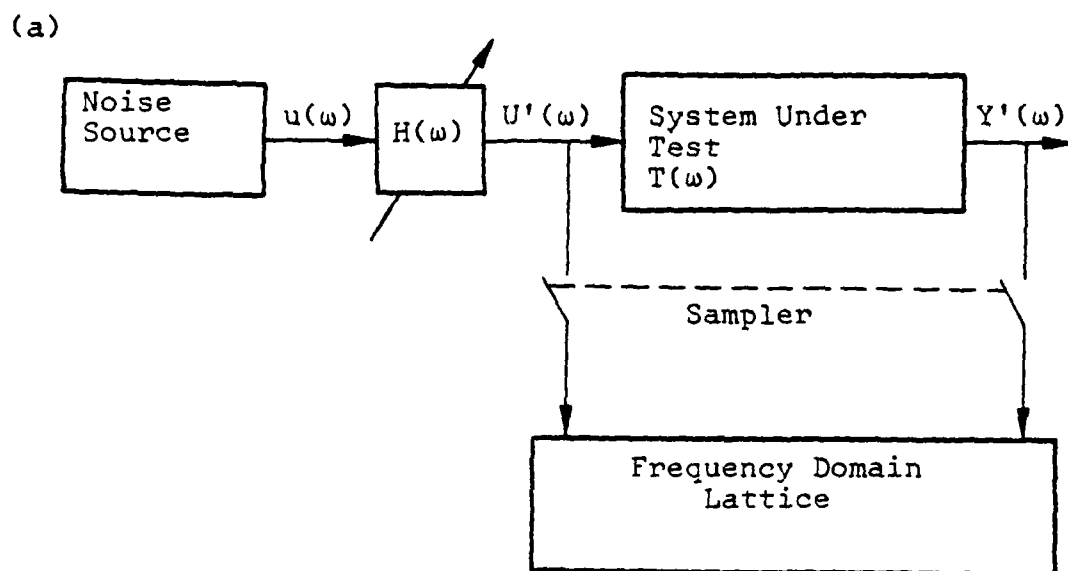


Figure 3.6 Lattice Input Preprocessing

(c)

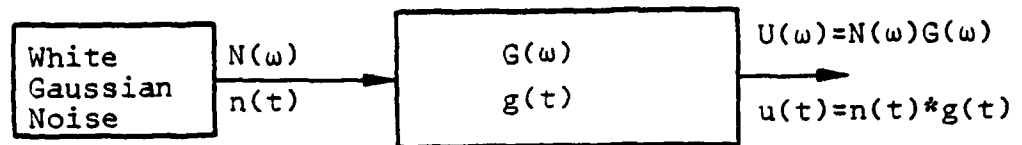


Figure 3.6 Lattice Input Preprocessing (con't)

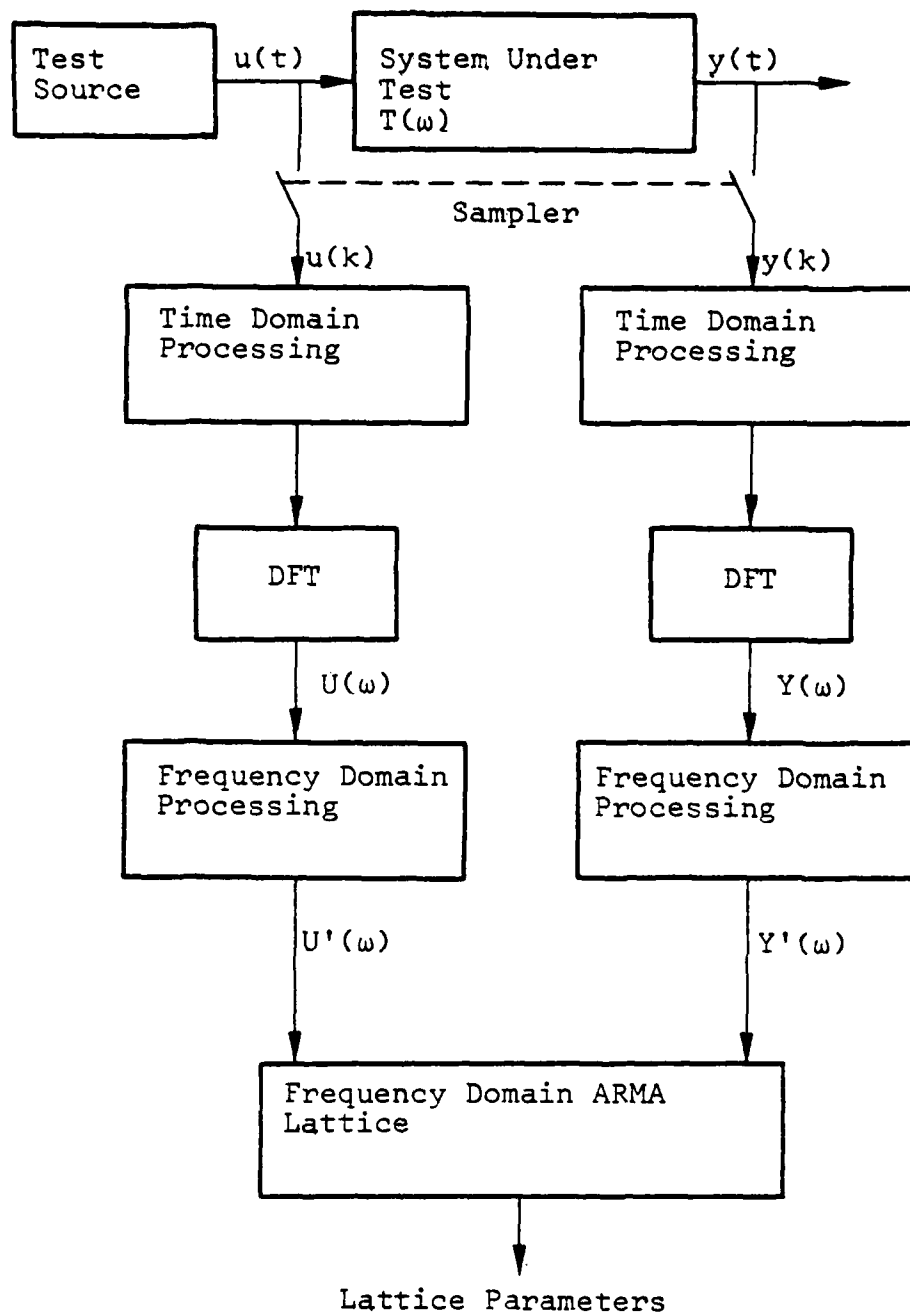


Figure 3.7 Frequency Domain ARMA Lattice with Preprocessing

A random noise generator can be represented as shown in Figure 3.6c. If this generator provides the source signal of Figure 3.7, then preprocessing to make $U'(\omega)$ white is equivalent to the multiplication in the frequency domain of $U(\omega)$ by $G^{-1}(\omega)$, assuming that the inverse exists and is stable.

In the discrete n -point frequency domain $U(\omega)$ can be written as a vector

$$\underline{U}^T = [U_0 \ U_1, \dots, U_{n-1}] \quad (3-74a)$$

$G(\omega)$ in its discrete form can also be written as a vector

$$\underline{G}^T = [G_0 \ G_1 \ \dots \ G_{n-1}] \quad (3-74b)$$

Multiplying by $G^{-1}(\omega)$ can be expressed in the discrete frequency domain as a multiplication of the vector \underline{U} by the square matrix \underline{H} , namely $\underline{U}'^T = \underline{U}^T \underline{H}$, where

$$\underline{H} = [h_{ij}] \quad \text{and} \quad h_{ij} = 0 \quad \text{for } i \neq j \quad (3-75a)$$

and

$$h_{ii} = \bar{U}_i^{-1} \quad \text{for } i = 1, 2, \dots, n \quad (3-75b)$$

where \bar{U}_i is the expected value of the i th harmonic of the input signal $U(\omega)$ taken over an ensemble of the input signals.

For \underline{H} to be useful $G(\omega)$ should have no poles or zeros on the unit circle. $G(\omega)$ can be known apriori (or estimated) from an autoregressive analysis of the signal generator output, or by averaging the spectra of several (possible overlapping) segments of the generator output. Equations (3-75a) and (3-75b) are useful when dealing with an ensemble of input signals. Alternately for a single member of the ensemble

$$h_{ii} = U_{i-1}^{-1} \text{ for } i = 1, 2, \dots, n \quad (3-75c)$$

can be used. This is equivalent in the time domain to exciting the system with a unit impulse.

From Figure 3.6c $U(\omega) = N(\omega)G(\omega)$. From Figure 3.6a the inputs to the lattice filter are given by

$$U'(\omega) = U(\omega)H(\omega) = N(\omega)G(\omega)H(\omega) = N(\omega)$$

when

$$H(\omega) = G^{-1}(\omega). \text{ Also,}$$

$$Y'(\omega) = U'(\omega)T(\omega) = N(\omega)T(\omega)$$

and the transfer function derived by the lattice model is

$$\frac{Y'(\omega)}{U'(\omega)} = T(\omega)$$

From Figure 3.6b $U'(\omega) = U(\omega)H(\omega)$ and $Y'(\omega) = Y(\omega)H(\omega) = U(\omega)T(\omega)H(\omega)$.

The transfer function for the lattice is

$$\frac{Y'(\omega)}{U'(\omega)} = T(\omega)$$

which is equivalent to that of Figure 3.6a. In Figure 3.6b the spectra presented to the lattice filter are not functions of $G(\omega)$, and the coefficients obtained can be used directly to identify the system under test.

E. FREQUENCY DOMAIN ARMA LATTICE SIMULATIONS

To verify some of these concepts a FORTRAN program was written that implements the ARMA frequency domain lattice with preprocessing introduced as shown in Figure 3.8. A Hamming window can be selected for preprocessing in the time domain. The selectable frequency domain processing implemented is that of (3-75c). The spectrum $Y(\omega)$ is divided by the spectrum $U(\omega)$ to give $Y'(\omega)$, the new system output spectrum. The input spectrum $U(\omega)$ is replaced with a unit magnitude real spectrum.

When preprocessing is not desired then the inputs to the ARMA frequency domain lattice are changed to $U'(n) = U(n)$ and $Y'(n) = Y(n)$.

For comparison a program was written that solved the ARMA normal equations by Gaussian elimination. Listings of these programs are available from the author.

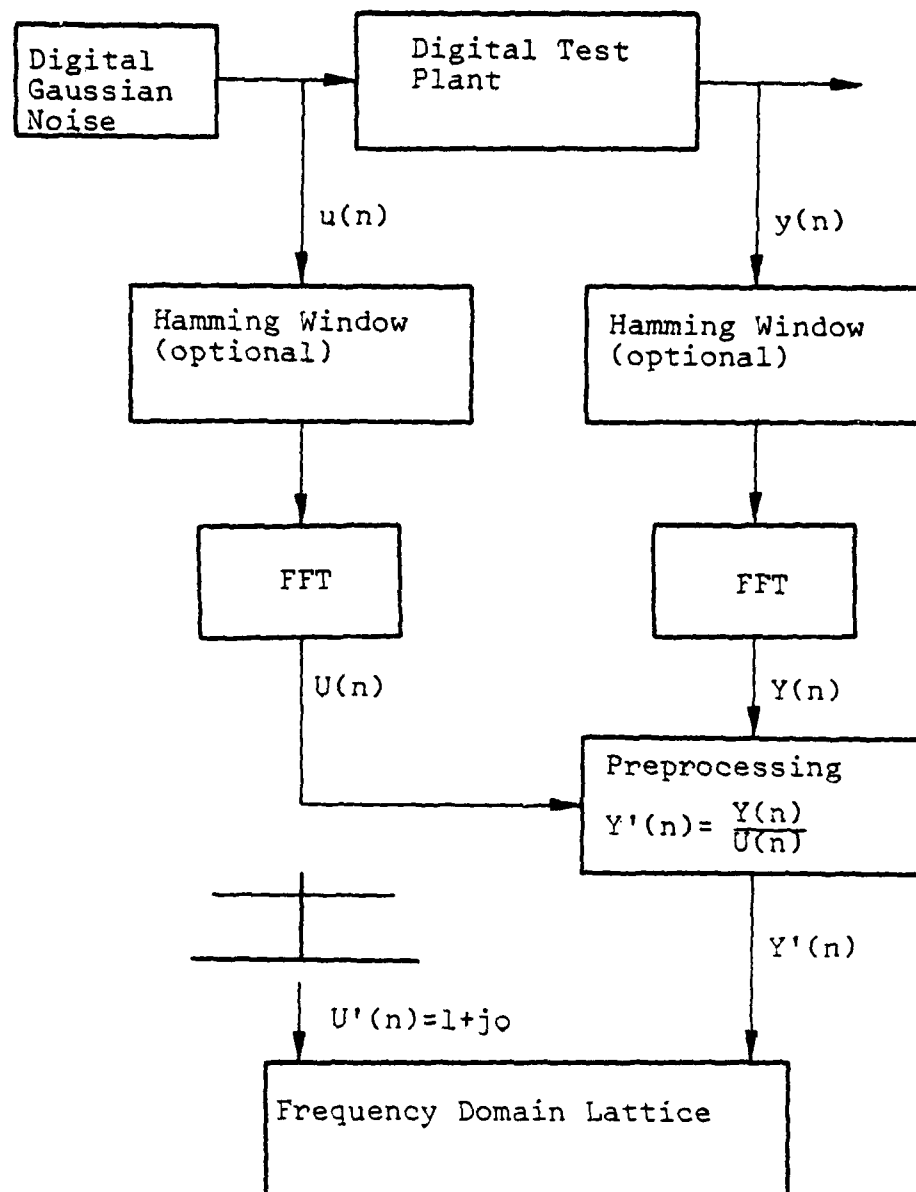


Figure 3.8 ARMA Lattice Simulation Program Block Diagram with Input Preprocessing

More than twenty different systems, of up to fourth order, were used as the system under test. The number of data points used for a single analysis varied from 32 to 4096. The number of points was always an integral power of two because of an FFT routine used in the frequency domain lattice.

To illustrate the ability of the frequency domain lattice to accurately model a system two examples, typical of all the problems exercised are shown here. System 1,

TABLE II
SYSTEM 1

Transfer Function Coefficients

Numerator		Denominator	
A(0) = 0.25		B(1) = -1.14	
A(1) = 0.35		B(2) = 1.4549	
A(2) = 0.245		B(3) = -0.849	
A(3) = 0.0		B(4) = 0.40745	
A(4) = 0.0			
Zeros		Poles	
Real	Imaginary	Real	Imaginary
-0.70	0.70	0.50	0.50
-0.70	-0.70	0.50	-0.50
		0.07	0.90
		0.07	-0.90

characterized in Table II, is a fourth order low pass system. Figure 3.9a and 3.9b show the magnitude of the transfer function of the model obtained by exciting the system with a single (256 point) sample of white noise from the Gaussian

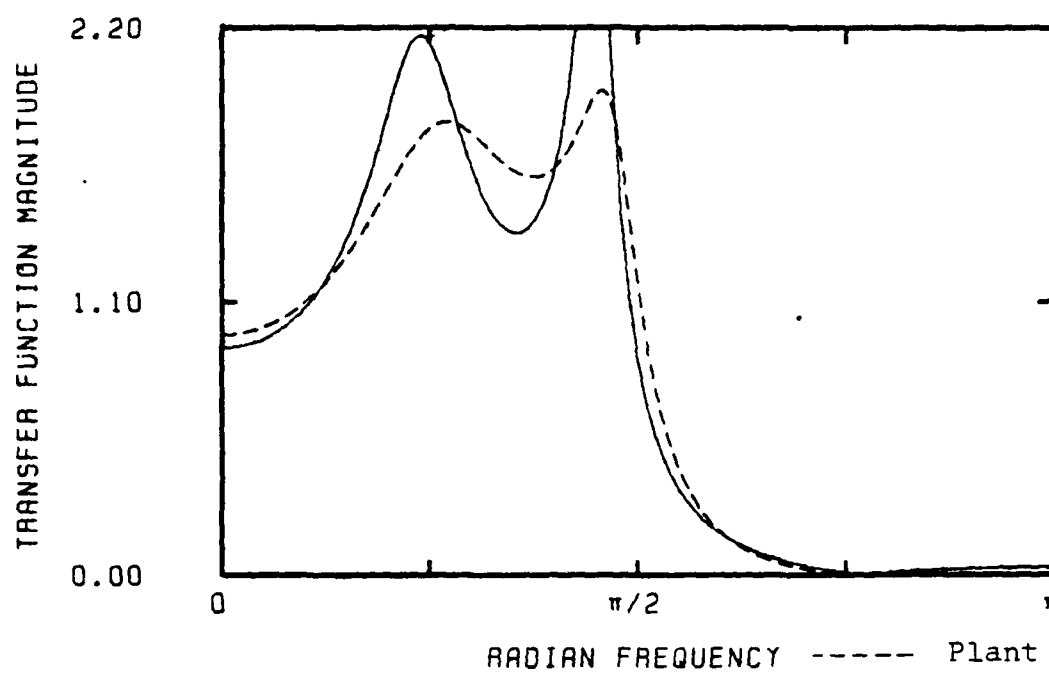


Figure 3.9a Gaussian Elimination 256 Points

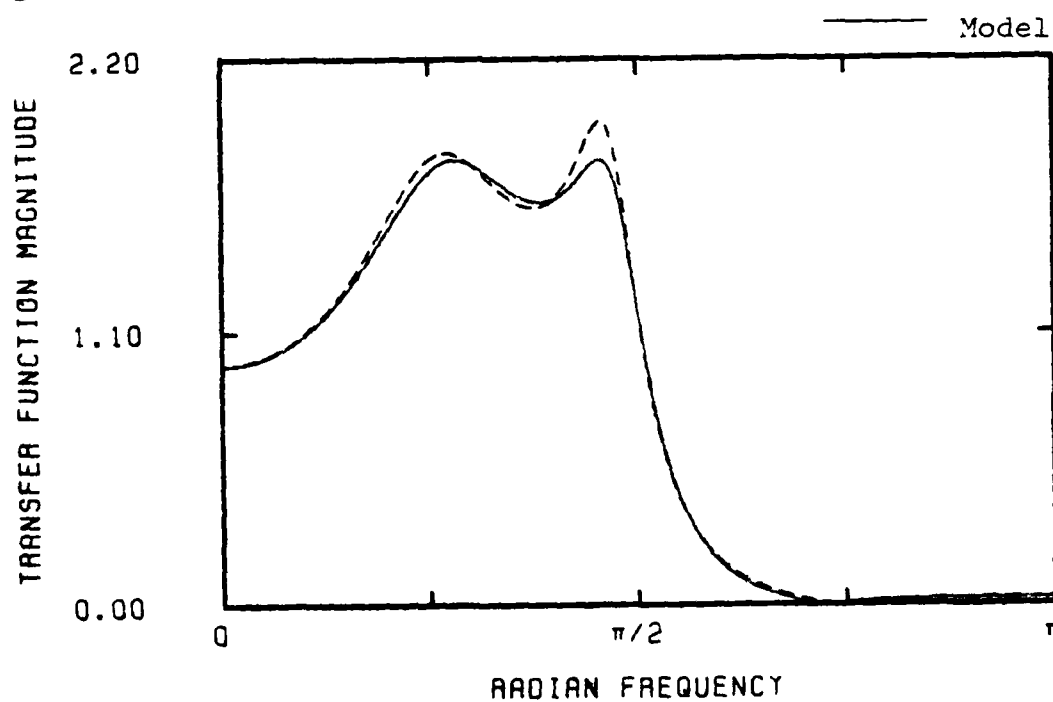


Figure 3.9b Lattice Method 256 Points

AD-A115 752

NAVAL POSTGRADUATE SCHOOL MONTEREY CA
EFFICIENT MULTICHANNEL AUTOREGRESSIVE MODELING IN TIME AND FREQ--ETC(U)
MAR 82 D J KLICH

F/G 12/1

UNCLASSIFIED

NL

2-2

2-2

2-2

END

DATE

FILED

7-82

DTIC

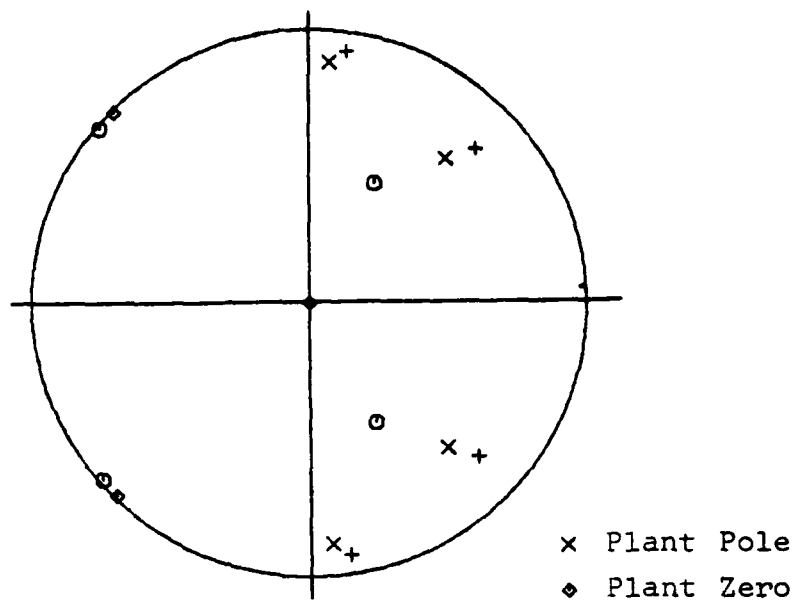


Figure 3.9c Gaussian Elimination 256 Points

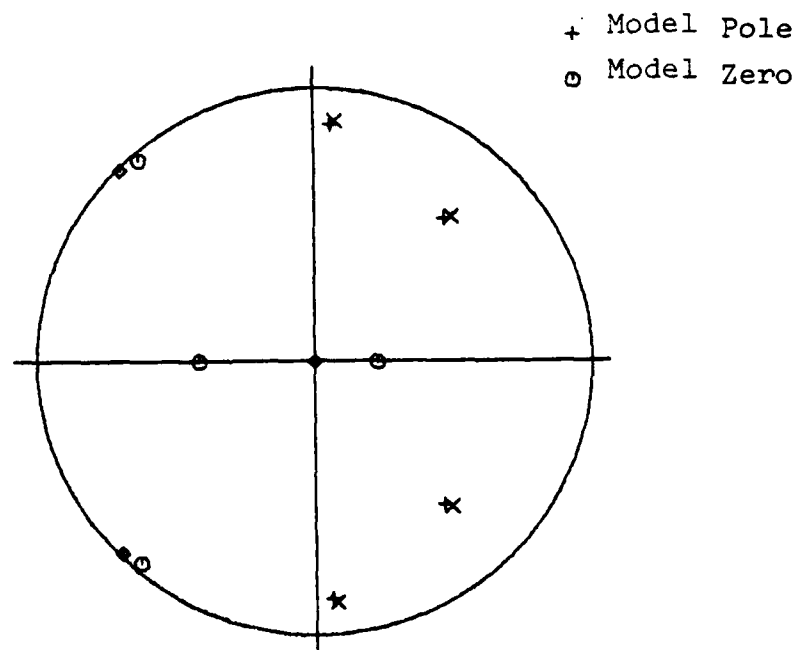


Figure 3.9d Lattice Method 256 Point

elements and the frequency domain lattice using no preprocessing. As has been observed in the past [15] the lattice results in a closer model of the system. Figure 3.9c and 3.9d show the pole zero plots of the resultant transfer functions.

Figure 3.10a and 3.10b are the transfer function magnitude found with the same methods using 4096 data points. Here the lattice model is still a more accurate model than the batch method.

While in most cases the lattice offered superior performance to the Gaussian elimination method, System 2, Table III, is a counter example that shows that this is not always the case. In this example the system is a second order band pass filter. Figure 3.11a and 3.11b are the transfer function magnitude plots found using 128 points, and Figure 3.11c and 3.11d show the pole zero plots. Here it is seen that the batch method produces a superior model. Figure 3.12 shows the same system analysis using 1024 points, and both methods accurately model the system.

To determine the ability of the input preprocessing to compensate for nonwhite input to the system the arrangement of Figure 3.13 was used to generate a colored input signal.

The plant used is that of System 1. The test input is a white Gaussian noise source followed by a single pole filter, with the pole position varied from -0.5 to 0.5 for different

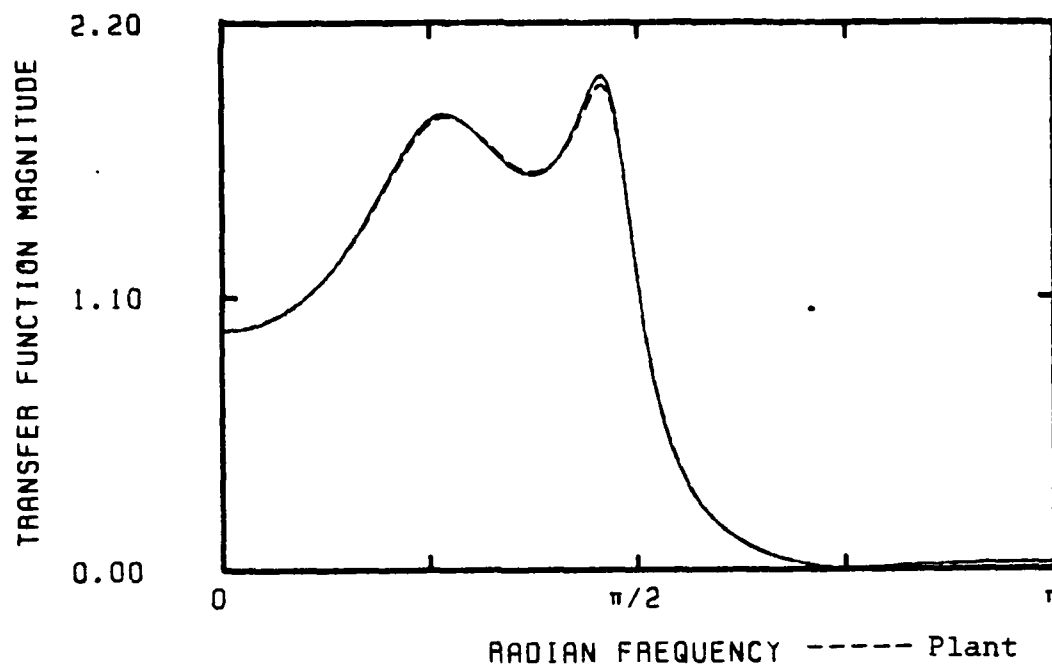


Figure 3.10a Gaussian Elimination 4096 Points

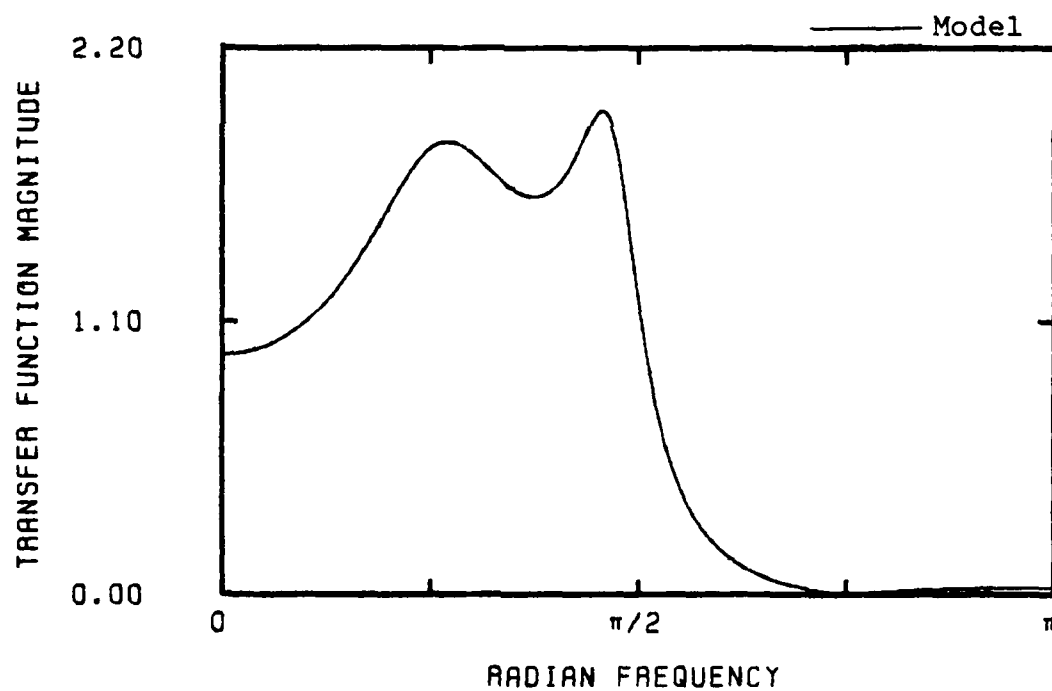


Figure 3.10b Lattice Method 4096 Points

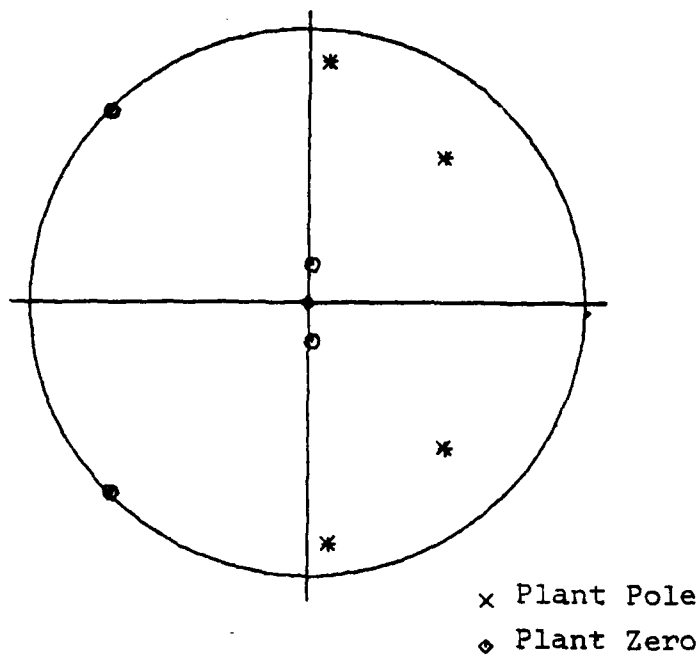


Figure 3.10c Gaussian Elimination 4096 Points

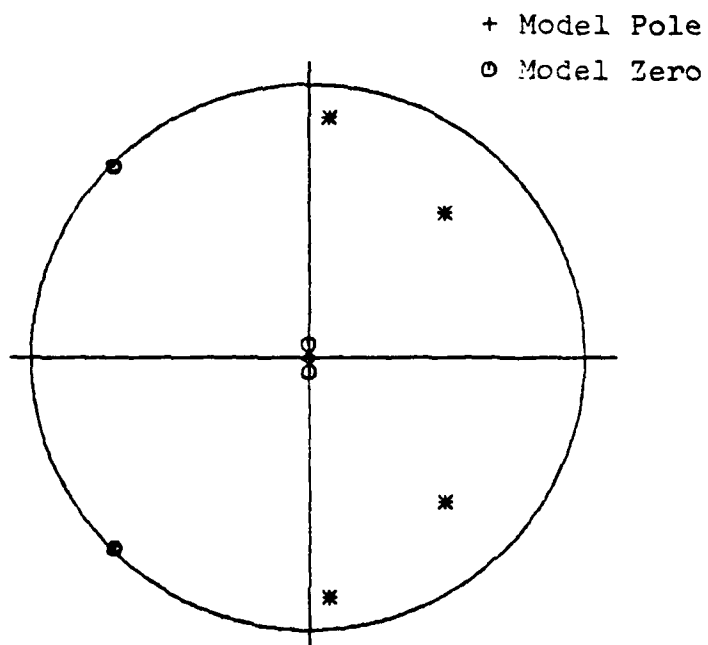


Figure 3.10d Lattice Model 4096 Points

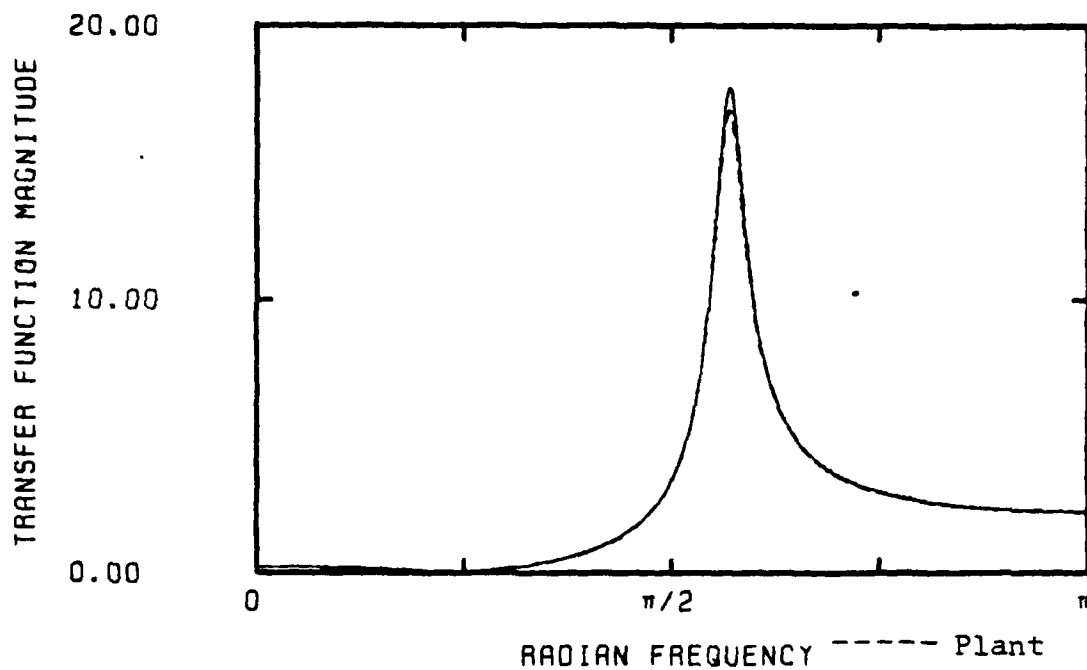


Figure 3.11a Gaussian Elimination 128 Points

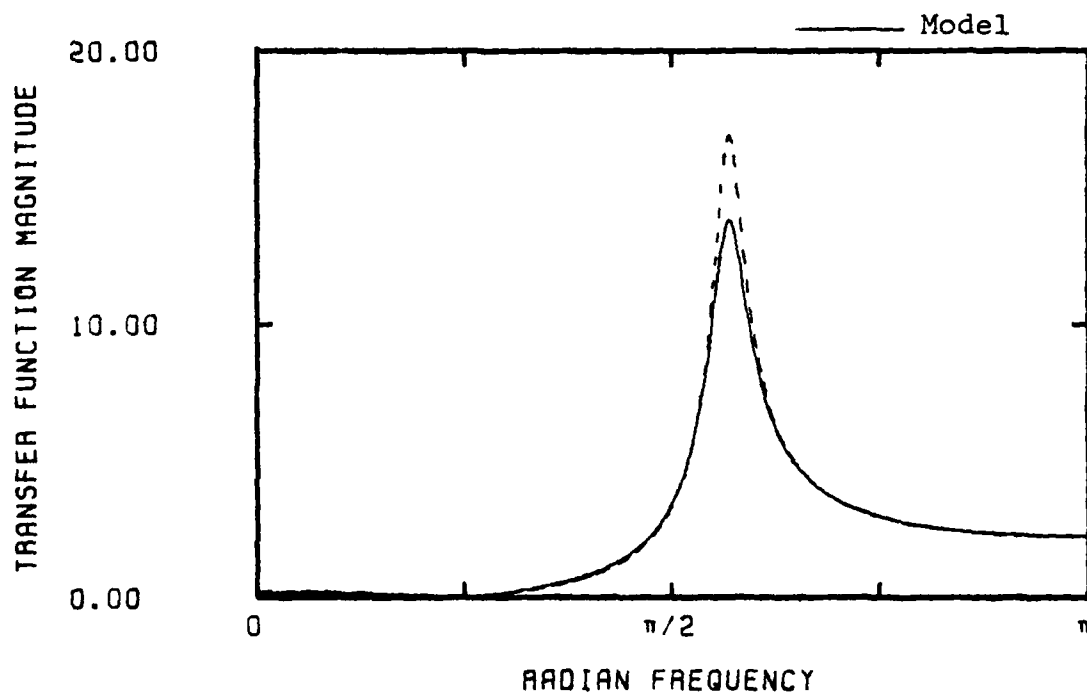


Figure 3.11b Lattice Method 128 Points

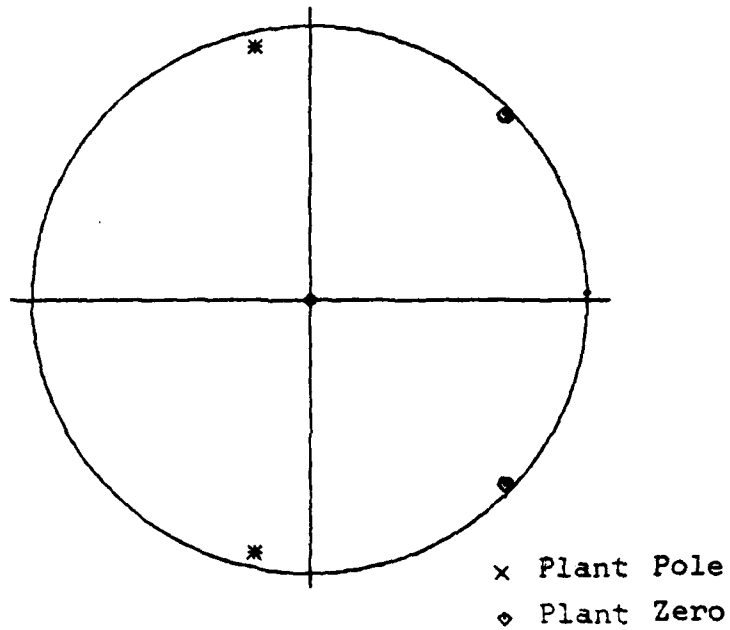


Figure 3.11c Gaussian Elimination 128 Point

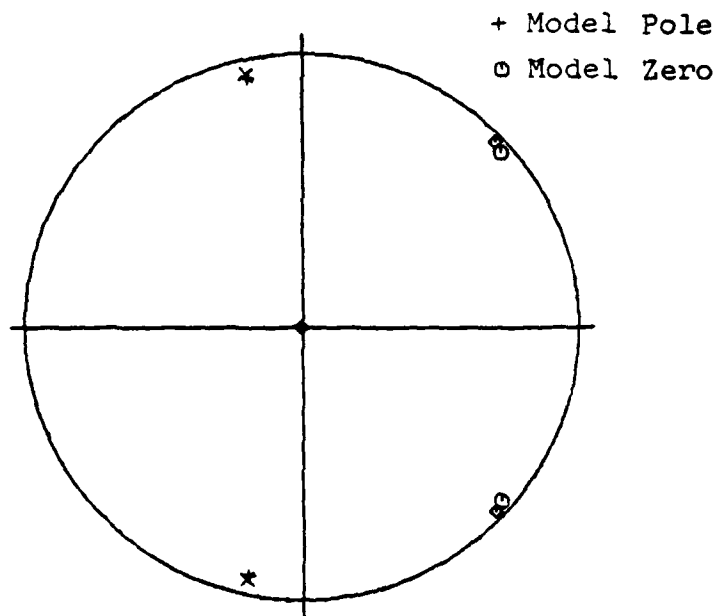


Figure 3.11d Lattice Method 128 Points

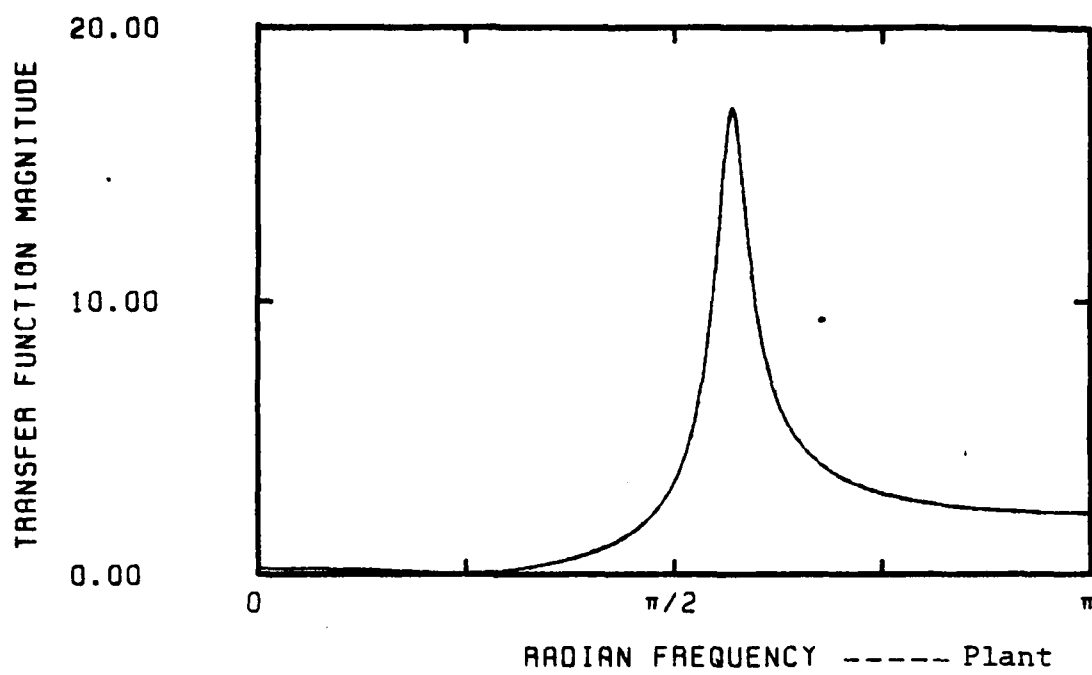


Figure 3.12a Gaussian Elimination 1024 Points

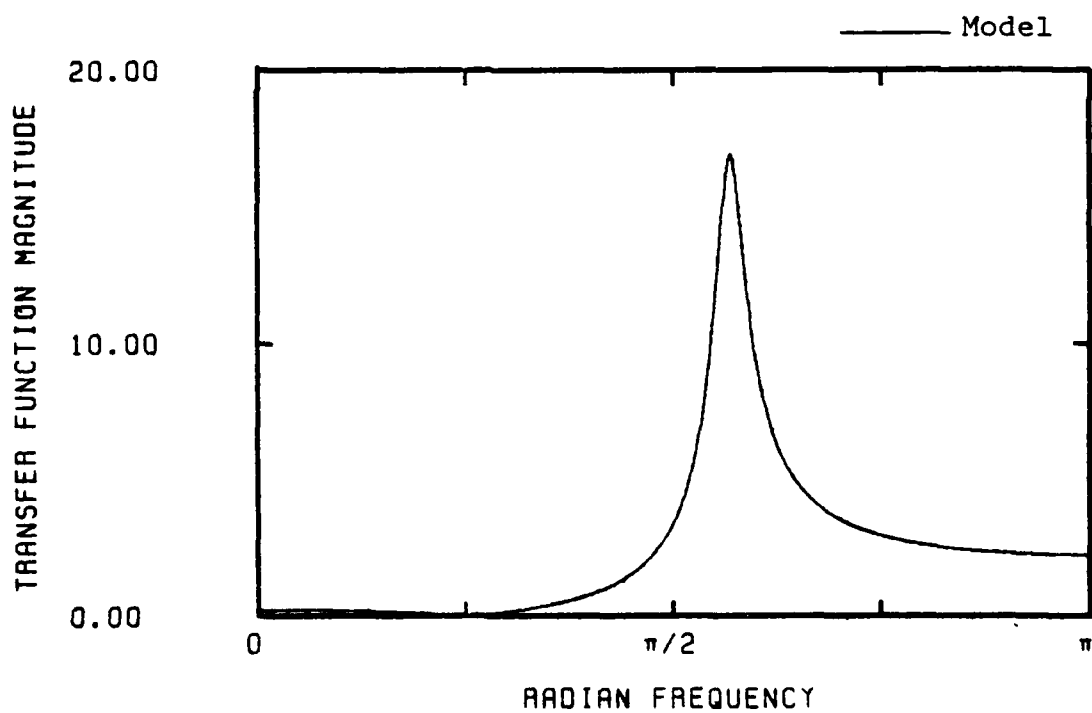


Figure 3.12b Lattice Method 1024 Points

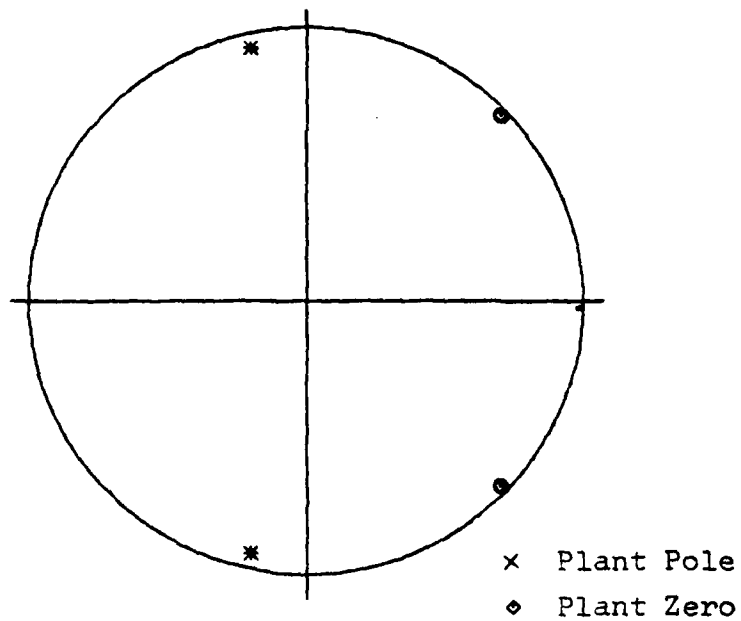


Figure 3.12c Gaussian Elimination 1024 Points

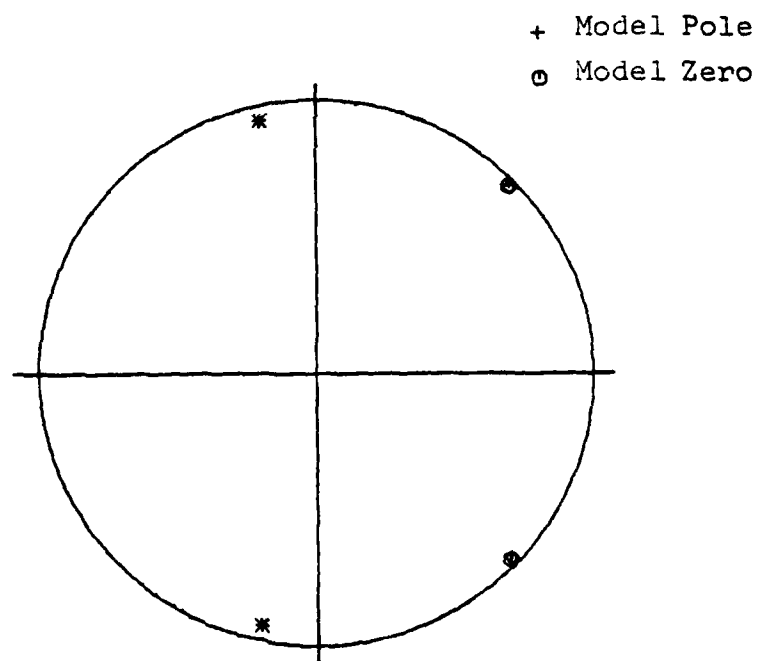


Figure 3.12d Lattice Method 1024 Points

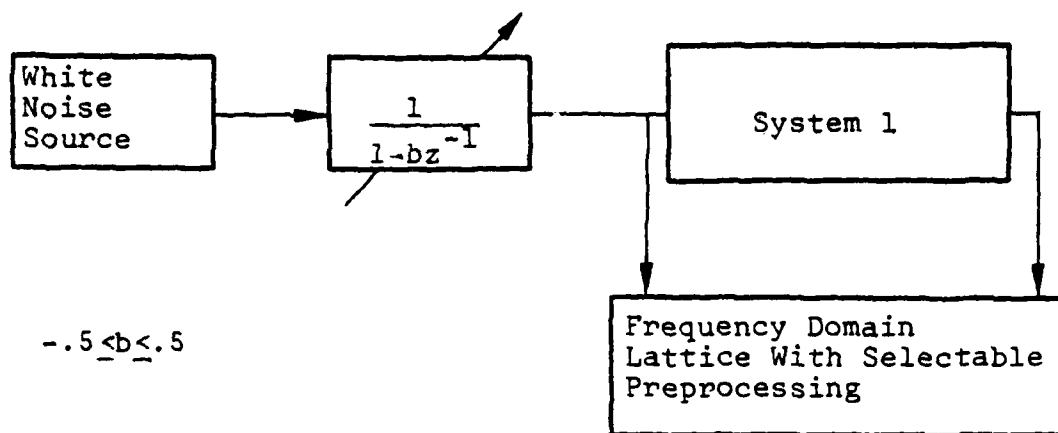


Figure 3.13 Input Preprocessing Test

TABLE III

SYSTEM 2

Transfer System Coefficients

Numerator		Denominator	
A(0) = 1.00		B(1) = 0.40	
A(1) = -1.40		B(2) = 0.89	
A(2) = 0.95			
Zeros		Poles	
Real	Imaginary	Real	Imaginary
0.700	0.67823	0.200	0.92195
0.700	-0.67823	0.200	-0.92195

test runs. The filter was operated under three conditions of preprocessing:

- 1) No preprocessing.
- 2) Inputs $U(\omega)$ and $Y(\omega)$ normalized as in (3-75c)
- 3) Multiplication of the input data by a Hamming window and then normalizing in the frequency domain as in (3-75c).

Figure 3.14 shows two typical plots of the first stage lattice parameters. The horizontal axis is the pole position, b , of the input filter of Figure 3.13. The vertical axis is the value of the lattice parameter, in this case $k_{22}^{(1)}$ and $\bar{k}_{22}^{(1)}$ for cases 1) and 2). For all first stage parameters, case 3) results in essentially identical results to case 2). It is clear that the preprocessing has made the lattice parameters independent of the input filter characteristics.

Figure 3.15 shows two similar plots for second stage parameters $\bar{k}_{11}^{(2)}$ and $\bar{k}_{12}^{(2)}$. Again the preprocessing is effective, and case 3) is essentially identical to case 2).

Figure 3.16 shows parameter $k_{11}^{(3)}$. The first plot is that of case 1) and 2). Here the parameter shows some sensitivity to the pole position, though much less than the unnormalized input signal. The second plot is for cases 1) and 3). The windowed and compensated lattice still shows no sensitivity to the input filter pole position. Of the eight parameters, this one showed the most variation with the pole position.

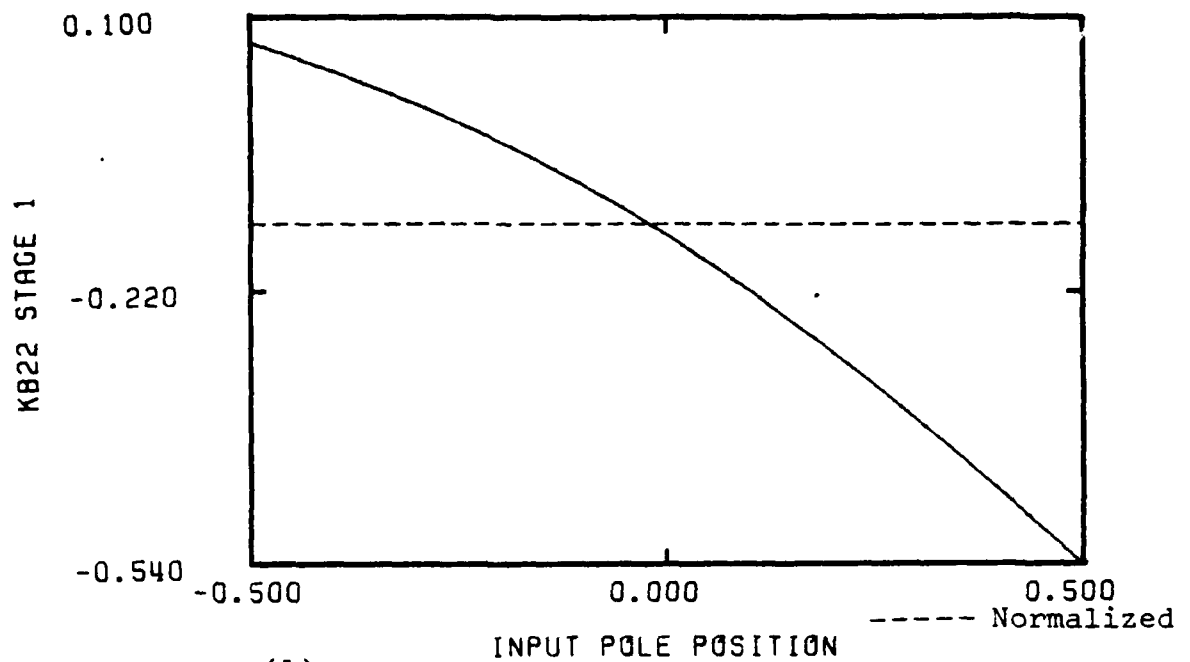


Figure 3.14a $\bar{k}_{22}^{(1)}$ vs. Input Pole Position

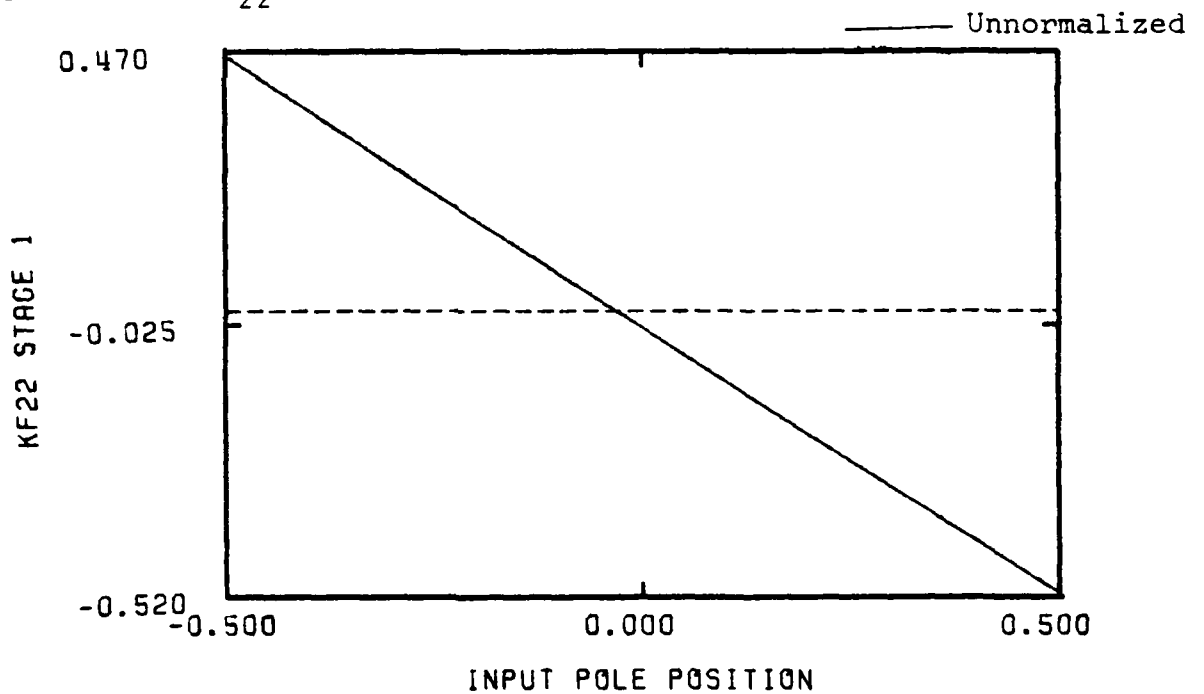


Figure 3.14b $k_{22}^{(1)}$ vs. Input Pole Position

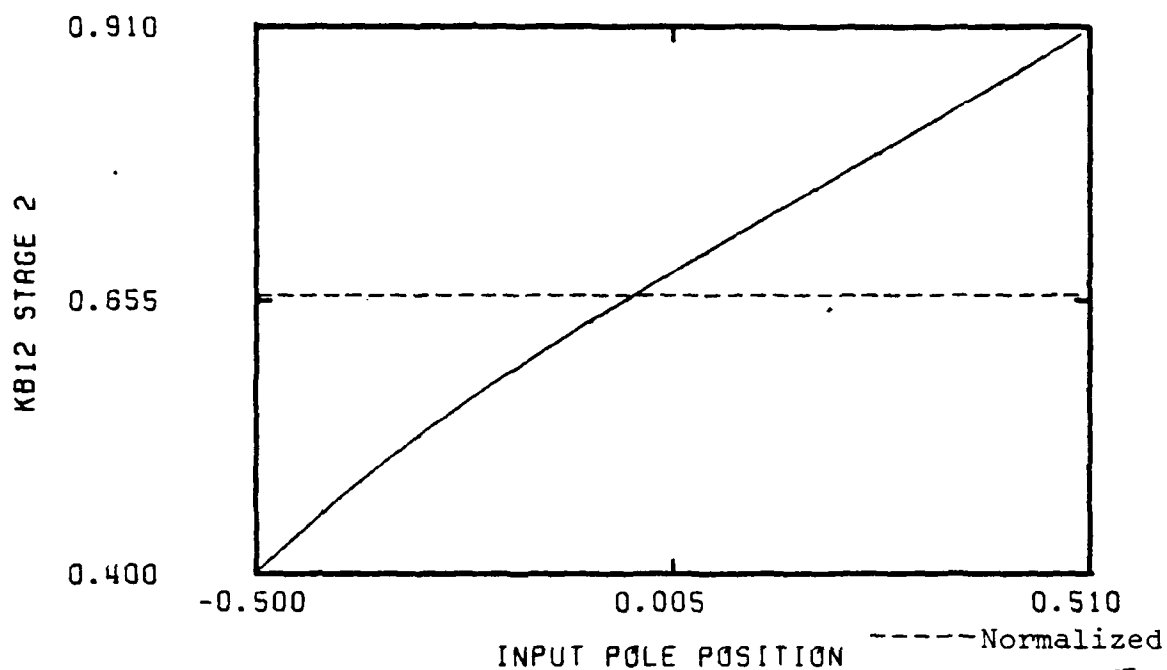


Figure 3.15a $\bar{K}_{12}^{(2)}$ vs. Input Pole Position

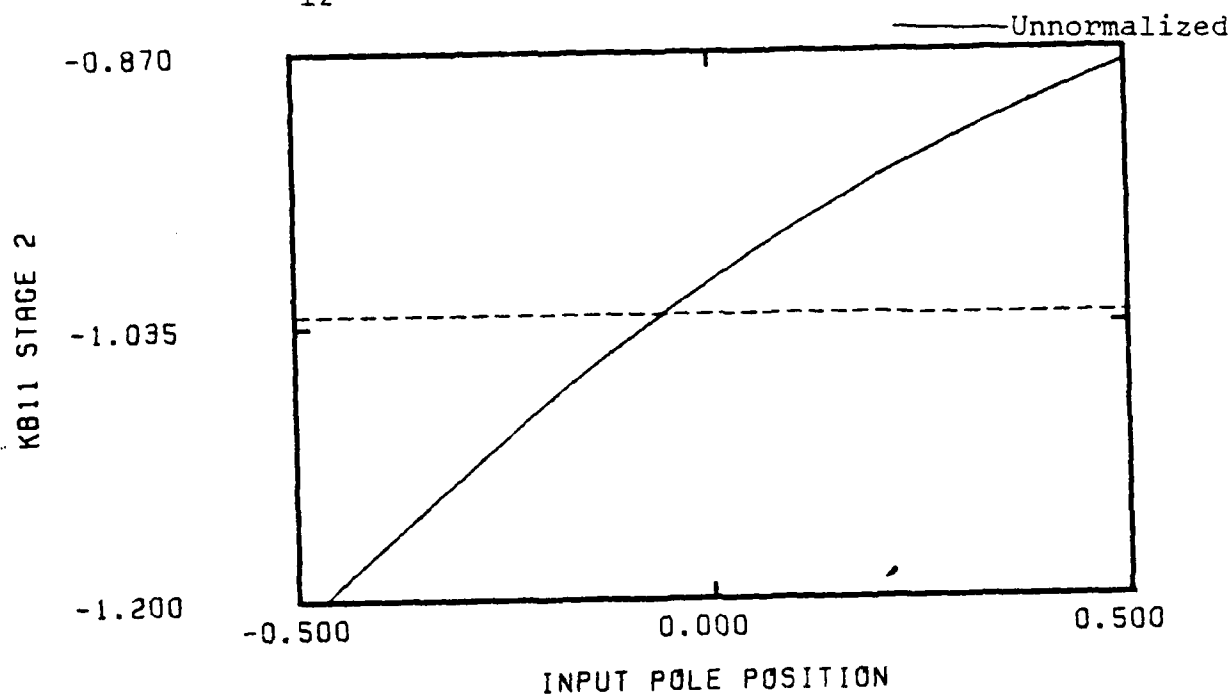


Figure 3.15b $\bar{K}_{11}^{(2)}$ vs. Input Pole Position

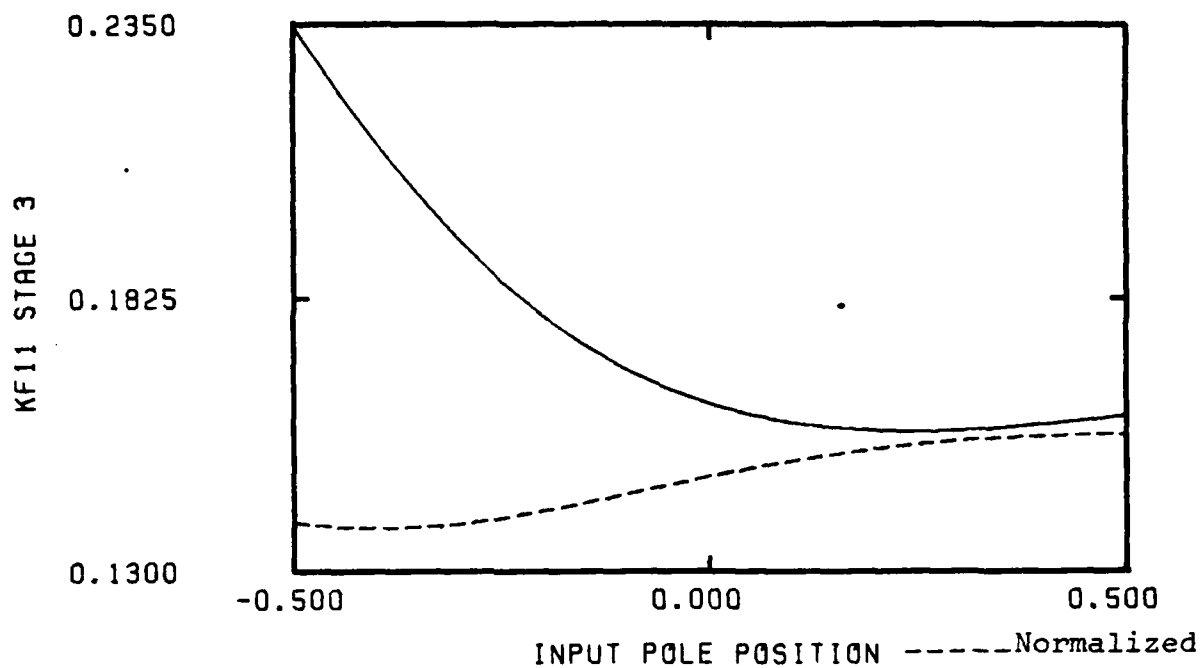


Figure 3.16a $k_{11}^{(3)}$ vs. Input Pole Position, Normalized

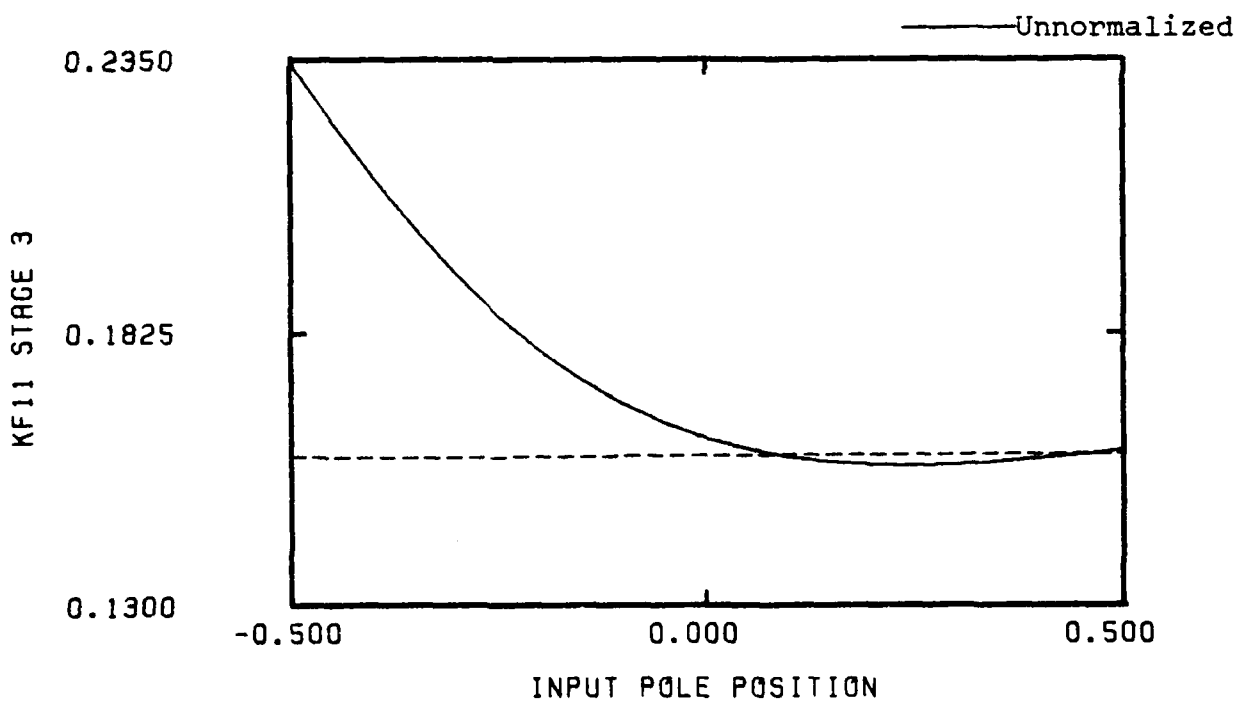


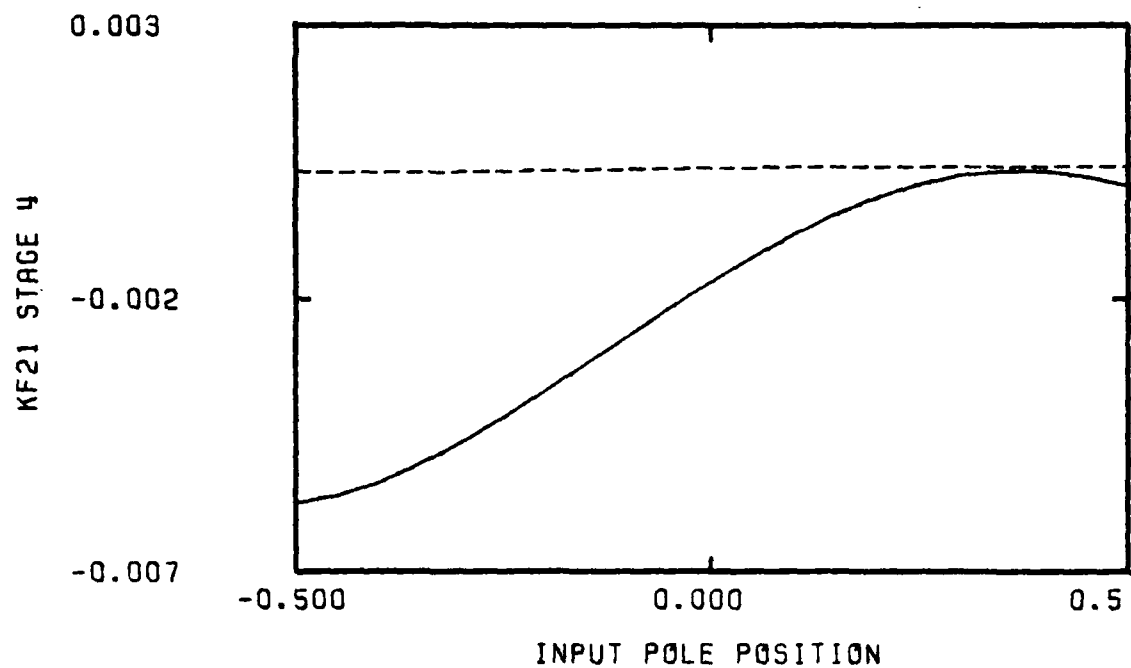
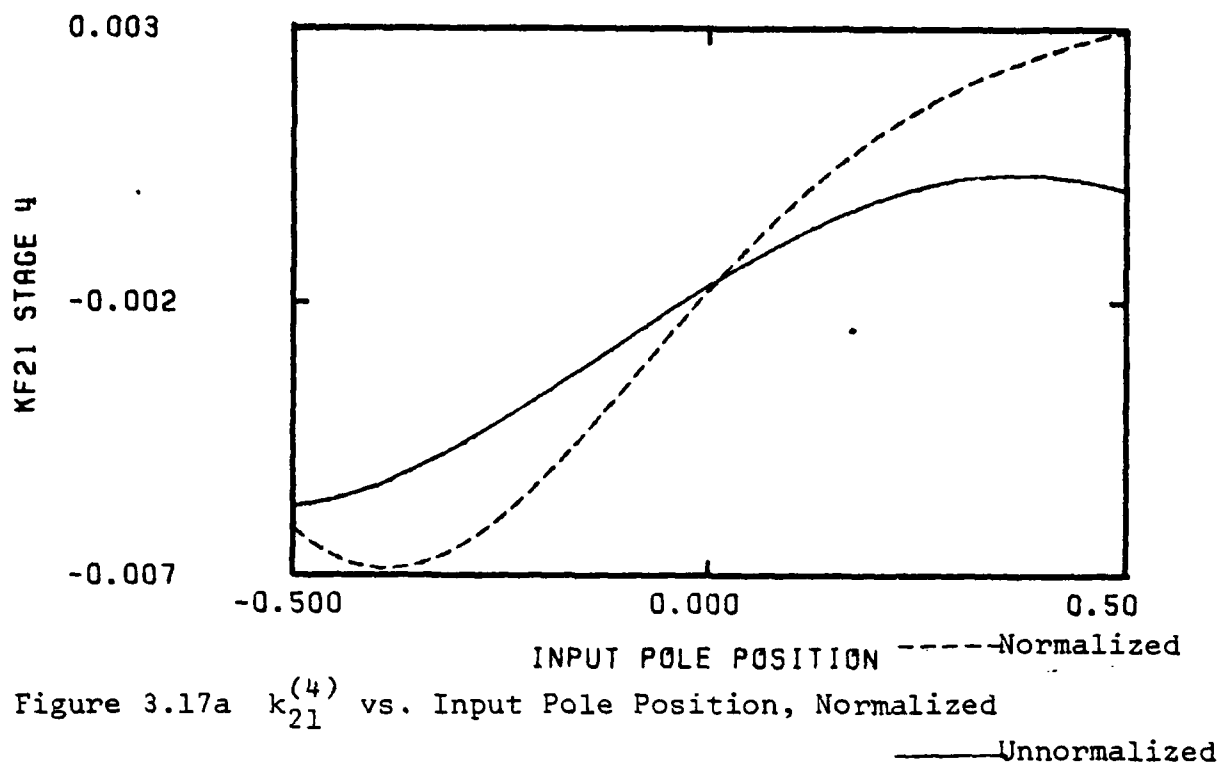
Figure 3.16b $k_{11}^{(3)}$ vs. Input Pole Position, Normalized and Windowed

Figure 3.17 is the fourth stage parameter $k_{12}^{(4)}$. Here the parameter calculated without windowing varies over an even wider range than the parameters calculated without any preprocessing. Several other of the parameters of the fourth stage do the same. The parameters of the windowed and compensated lattice, case 3), are all still insensitive to the input pole position.

It is apparent from this experiment that input signal preprocessing is effective in compensating for the effects of a nonwhite input signal, and results in a consistent set of lattice parameters for identification. The preprocessing should consist of windowing of the input data (to eliminate the Gibb's phenomenon) and normalization to whiten the signal.

F. FILTER SYNTHESIS WITH THE ARMA FREQUENCY DOMAIN LATTICE

So far the lattice filters have been developed as models for systems with measurable input and output. It is possible to define the input signal to the filter as a unit impulse so that the lattice models a desired response in the frequency or time domain. In this application the recursive in order nature of the lattice solution becomes of interest. The lattice can be expanded in order until a suitable fit of the desired specification is obtained.



Consider what happens when the frequency domain ARMA lattice is used to model a system when the excitation of that system is a unit impulse. The configuration is shown in Figure 3.18. $U(n)$, the spectrum of the input to the system under test, is a flat spectrum of unit magnitude, exhibiting the previously discussed desirable property of equal energy across the spectrum. The output spectrum of the system $Y(n)$ is exactly $H(n)$, the discrete Fourier transform of the impulse response of the system under test.

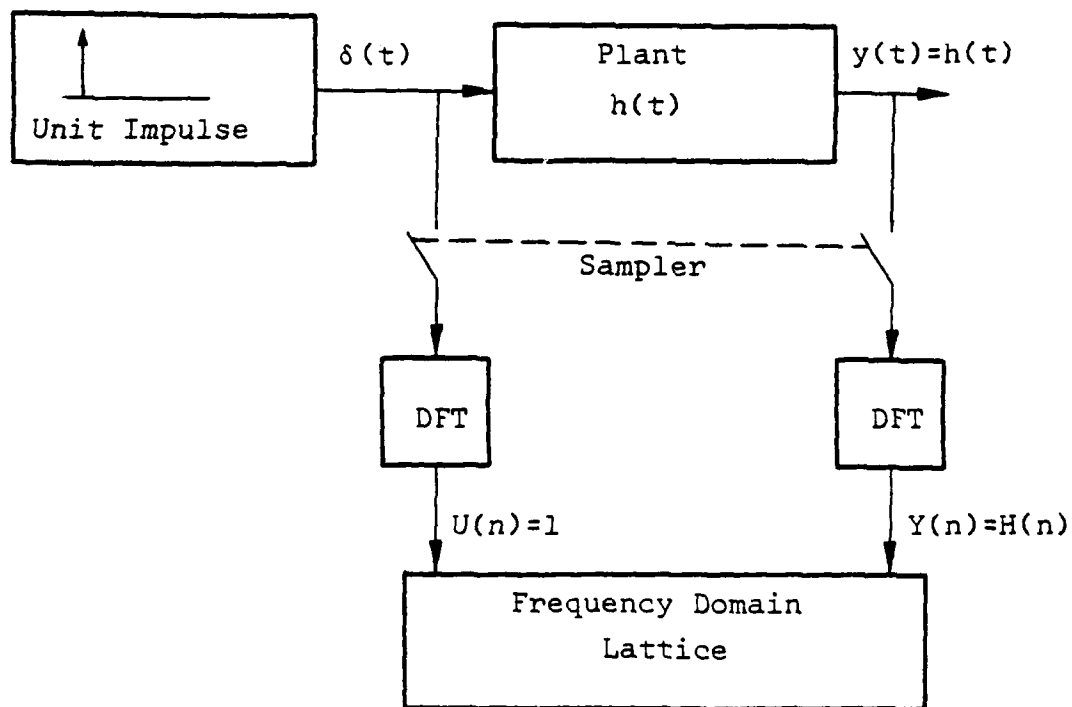


Figure 3.18 Frequency Domain Modeling

The lattice parameters obtained as a result of this analysis could be used to construct a synthesis model. This synthesis model would exhibit the best fit, in a least mean square sense, that a filter of its order could obtain. The order of the lattice can be increased to obtain an arbitrarily small MSE. When designing to a specification it is often desired that some portions of the frequency characteristics of the resultant filter be weighted more heavily than other. This too can be accomplished with the ARMA lattice. $U(n)$ and $Y(n)$ can both be multiplied by a weighting function which provides preemphasis to those portions of the spectrum for which a more accurate fit is desired.

1. Filter Synthesis Examples

To illustrate this capability it was postulated that a digital filter was desired with a magnitude

$$|H(n)| = 0.495 \cos \left(\frac{2\pi}{1024} \cdot n \right) + 0.505, \quad 0 \leq n \leq 1023$$

This desired magnitude was transformed to a minimum phase spectrum using a digital approximation of the Hilbert transform [14]. This spectrum was used as the input $Y(n)$ to the frequency domain lattice, with a unit real spectrum applied as the $U(n)$ input. The magnitude of the first and second order models of this desired spectrum are shown in Figure 3.19. The second order model magnitude appears to match

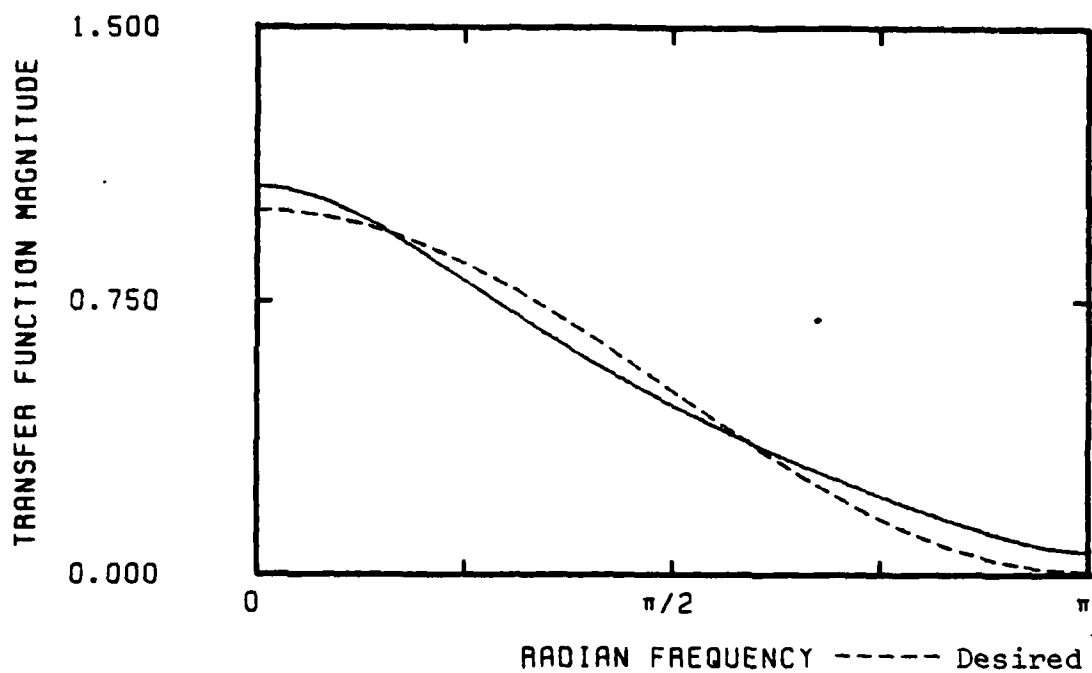


Figure 3.19a First Order Synthesis Model

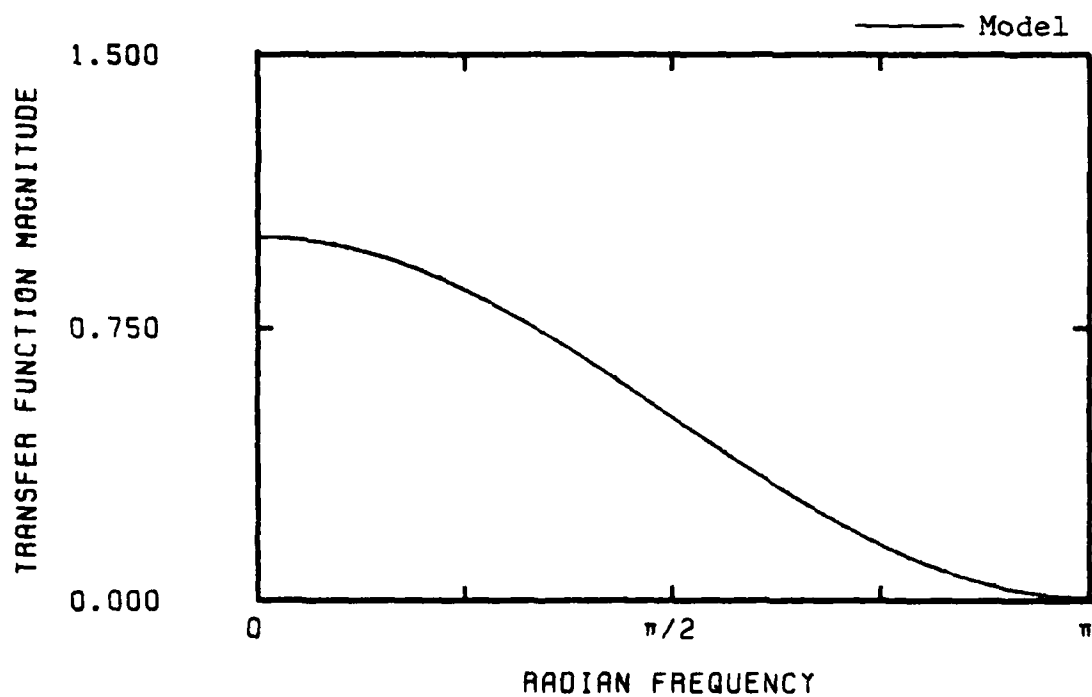
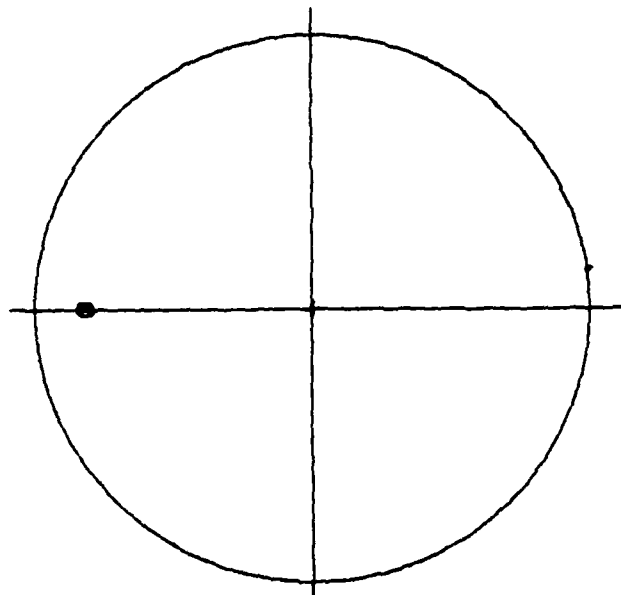


Figure 3.19b Second Order Synthesis Model



+ Model Pole
o Model Zero

Figure 3.19c Second Order Pole-Zero Plot

the desired magnitude, and as shown by the pole zero plot, Figure 3.19c, is stable and minimum phase. The digital transfer function of this second order filter is

$$H(z) = \frac{0.302504 + 0.495032z^{-1} + 0.202512z^{-2}}{1 + 0.00004z^{-1} + 0.000005z^{-2}}$$

As another example the frequency domain lattice is used to find a digital equivalent of the analog low pass circuit of Figure 3.20 which exhibits 40 dB attenuation at 5 kHz. This circuit was analysed using the SPICE program, a circuit analysis program. This program produced an AC analysis of the circuit, showing the complex output when the circuit is driven by a unit magnitude signal for frequencies from DC to 10 kHz, the equivalent of a Bode plot of the circuit transfer function.

This complex spectrum was applied to the frequency domain lattice as in the previous example. The third, fourth and fifth order solutions are shown in Figure 3.21. The transfer function of the fifth order solution is

$$H(z) = \frac{-0.00469 - 0.0269z^{-1} - 0.310z^{-2} - 0.03164z^{-3} - 0.316z^{-4} + 0.00096z^{-5}}{1 - 2.159z^{-1} + 2.534z^{-2} - 1.675z^{-3} + 0.6178z^{-4} - 0.0979z^{-5}}$$

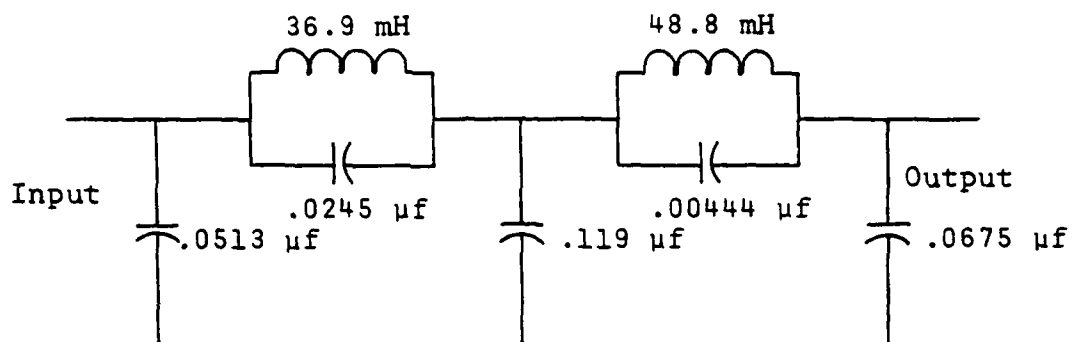
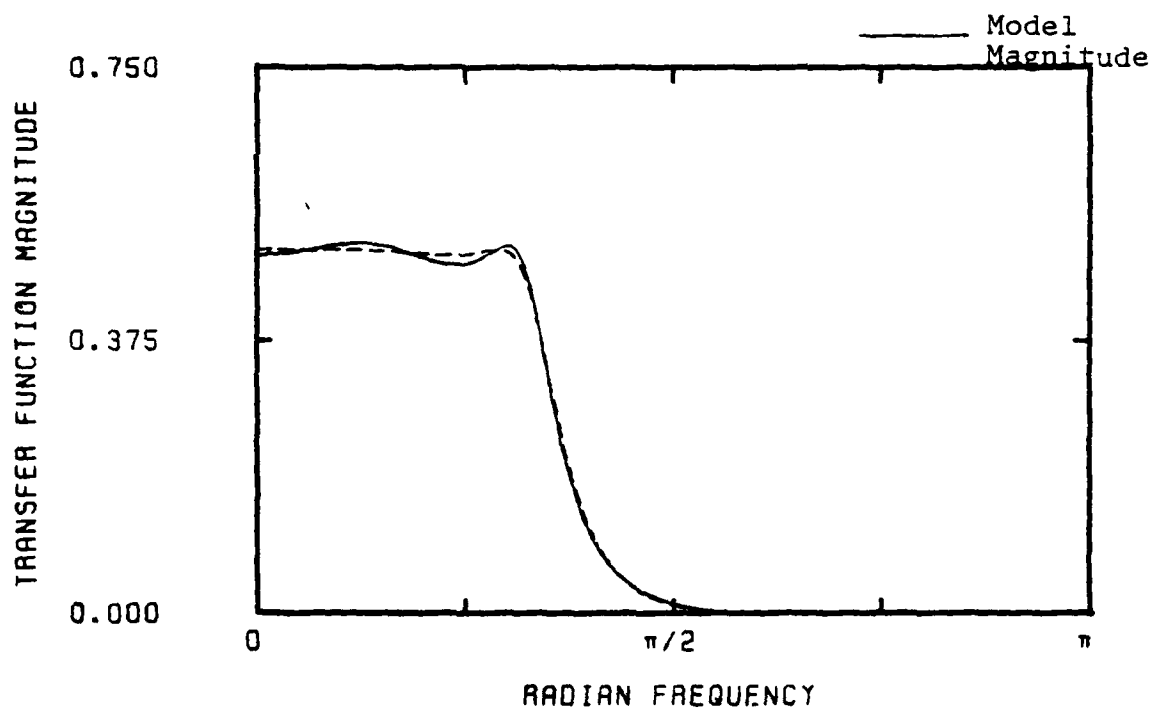
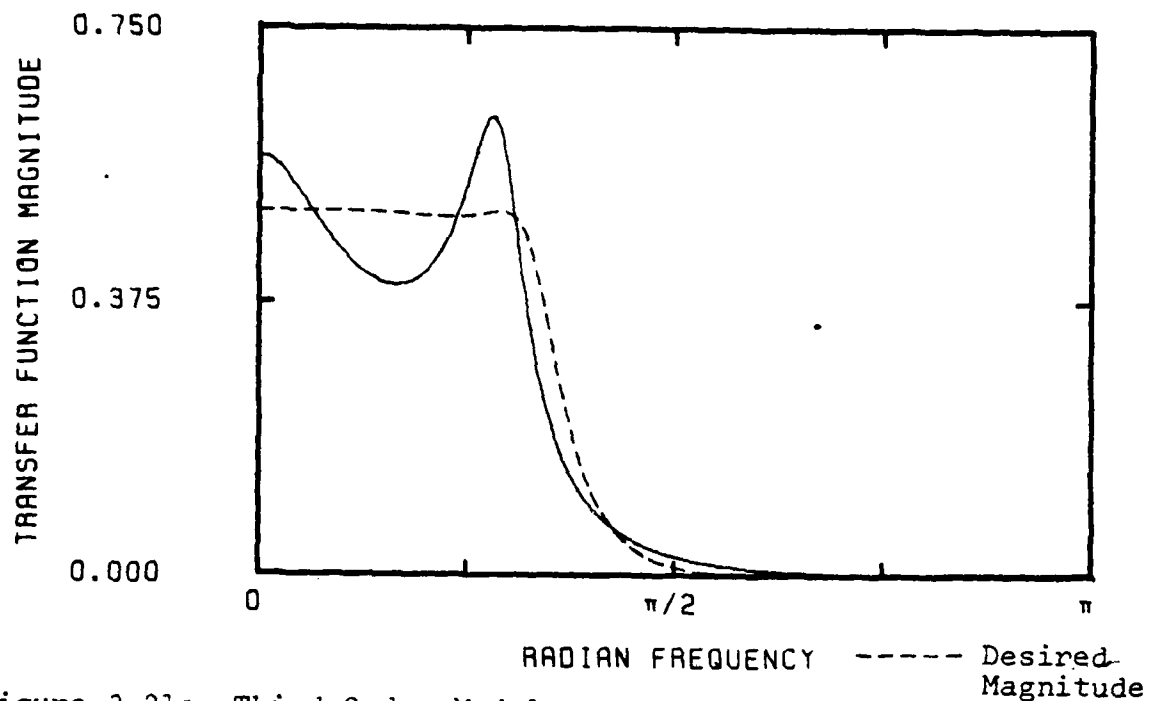


Figure 3.20 Analog Low Pass Filter

The pole zero and phase plots of the fifth order solution are shown in Figure 3.21. There is an additional pole not shown on the plot that is located at $z = -4.5$. The phase plot, Figure 3.21e, shows that the analog circuit has a phase at the Nyquist frequency of approximately $\pi/4$. Of course the digital filter cannot match this phase, it must be either zero or π at the folding frequency. It is believed that this discontinuity in the phase is the cause of the digital filters non minimum phase characteristic. The filter is stable, however, and does meet the desired magnitude specification.



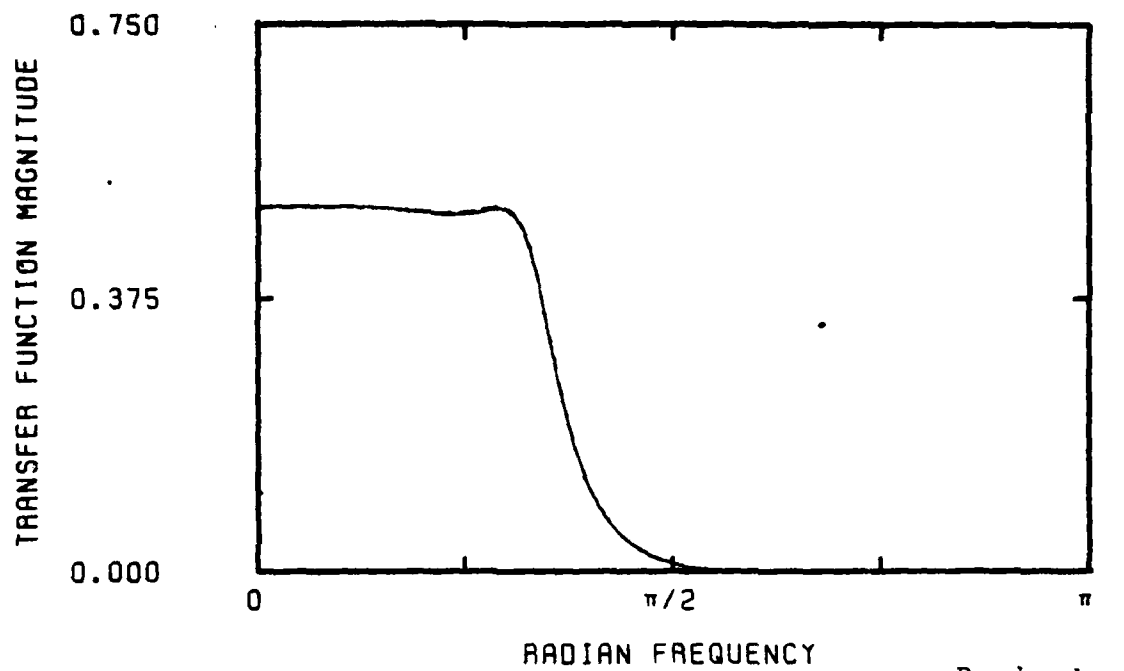


Figure 3.21c Fifth Order Model

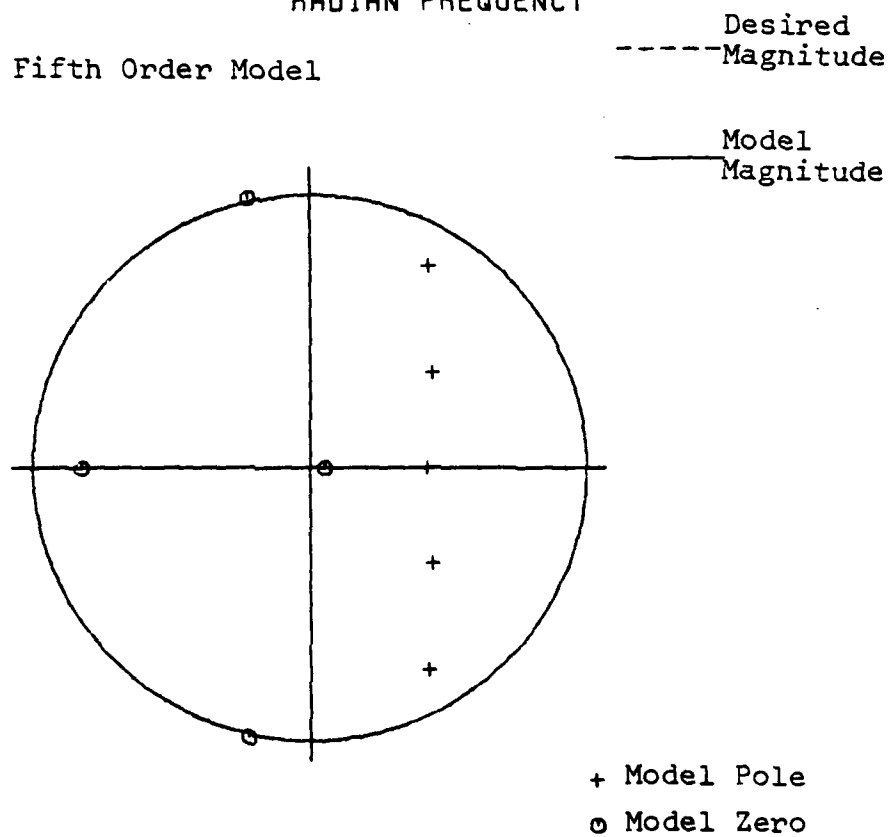


Figure 3.21d Fifth Order Model Pole-Zero Plot

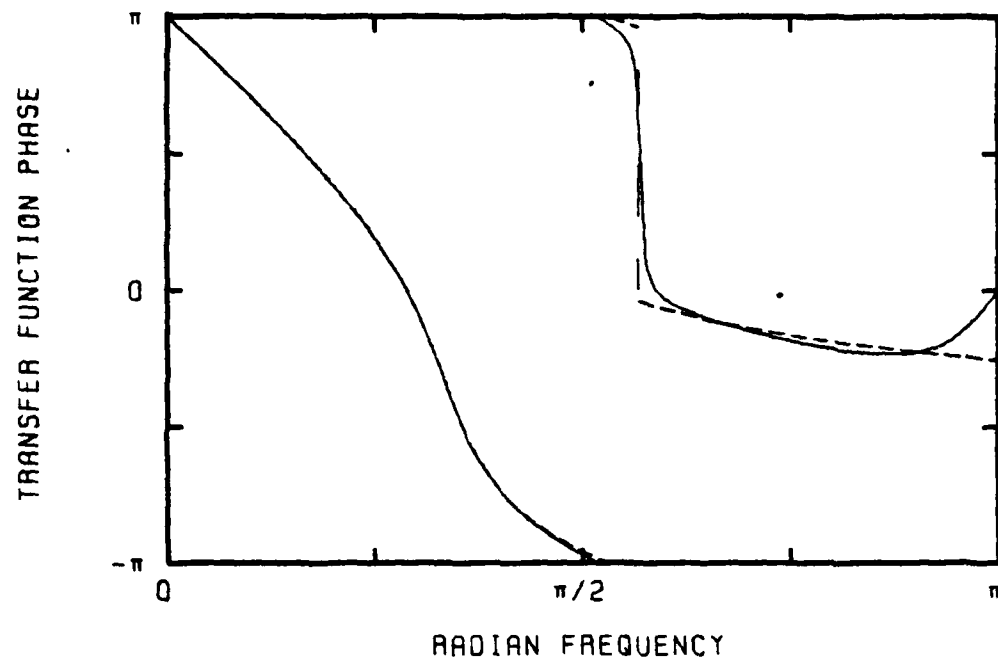


Figure 3.21e Fifth Order Model Phase

IV. AUTOREGRESSIVE LATTICE PARAMETERS CALCULATED USING THE CORRELATION MATRIX

In the foregoing chapters several methods of solution of the AR and ARMA normal equations have been discussed. These equations can be solved by matrix methods but for large order problems this is a computationally expensive method. If the available data is windowed to ensure a Toeplitz and symmetric correlation matrix the equations can be more efficiently solved using Levinson's algorithm. From Levinson's recursion the lattice structure can be implemented but requires more calculations because it is necessary to update prediction error signals at each stage. Makhoul [10] developed an algorithm that implements the single channel AR lattice but does not require the generation of these error signals, and thus is more efficient than other methods.

In this chapter a new implementation is developed for both the single channel and the two channel AR lattice that implements the lattice structure without the expense of calculating error signals. In the final formulation there is no requirement that the correlation matrix be Toeplitz or symmetric, removing constraints on how data is windowed and allowing solution of the problem when the system under analysis is not stationary. However, since the formal proof

depends upon the lattice formulation which assumes symmetric Toeplitz matrices, two solutions are presented, one assuming Toeplitz symmetric correlation matrices, the other omitting this restriction. This algorithm maintains the lattice advantages of maximum entropy and robustness that have made the lattice an attractive analysis structure. It requires less storage than conventional lattice implementations and can be extended efficiently to multichannel structures as are used for nonlinear systems analysis.

A. THE SINGLE CHANNEL AUTOREGRESSIVE LATTICE

In the single channel autoregressive lattice structure of Figure 4.1 the forward and backward error signals for each stage are updated using the equations from Section II.B.

$$e^{(n+1)}(k) = e^{(n)}(k) - k^{(n+1)} \bar{e}^{(n)}(k-1) \quad (4-1)$$

$$\bar{e}^{(n+1)}(k) = \bar{e}^{(n)}(k-1) - k^{(n+1)} e^{(n)}(k) \quad (4-2)$$

The reflection factors $k^{(n)}$ can be found from the backward equation (2-28)

$$k_B^{(n+1)} = \frac{\epsilon \{e^{(n)}(k) \bar{e}^{(n)}(k-1)\}}{\epsilon \{e^{(n)}(k) e^{(n)}(k)\}} \quad (4-3)$$

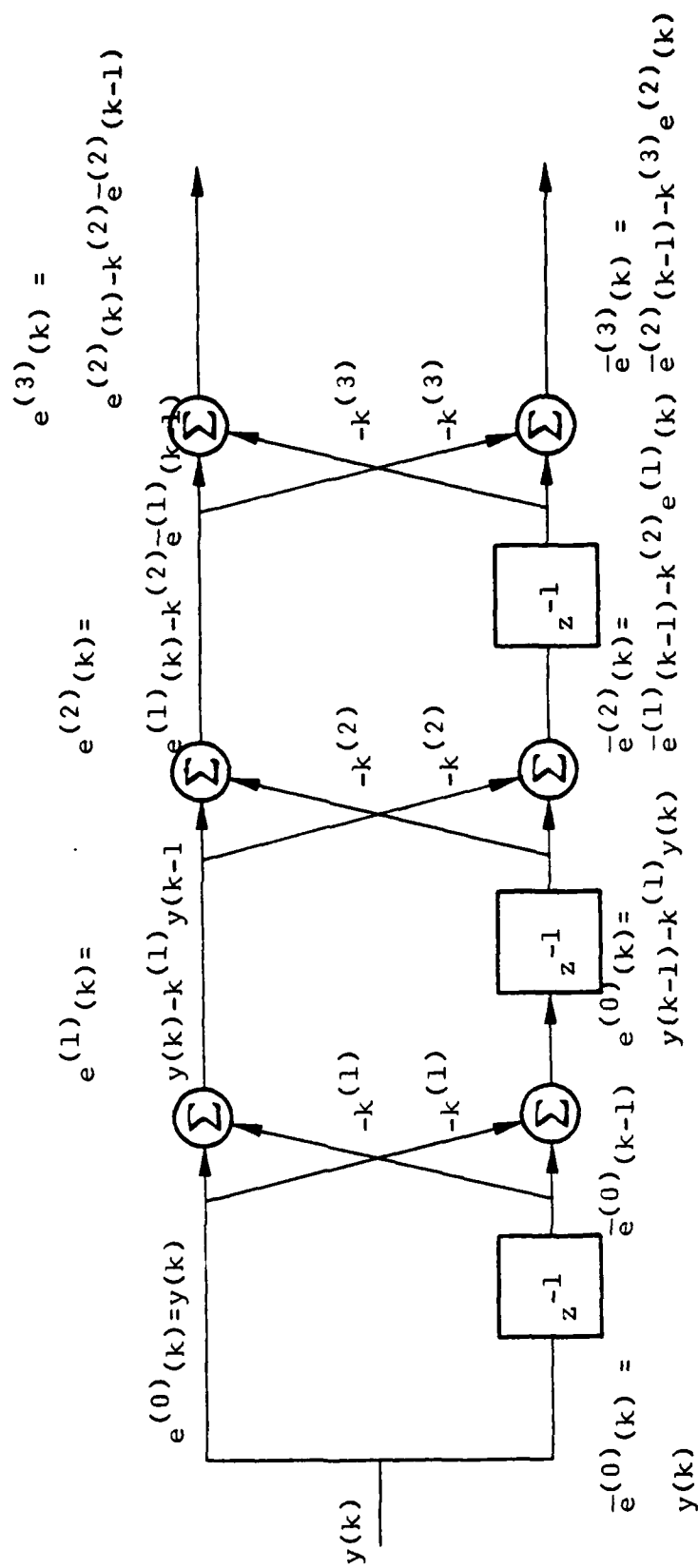


Figure 4.1 Single Channel Lattice Structure

Since the quantities $e^{(n)}(k)$ and $\bar{e}^{(n)}(k)$ are sums of weighted present and past values of the input $y(k)$, the numerator and denominator of (4-3) can be expressed as the weighted sum of correlations of the input vector \underline{y} . An algorithm based upon this observation can be expected to be computationally more efficient than previous algorithms if the correlations of the input signal are known since it is not necessary to update the error sequences at each lattice stage.

From Figure 4.1 the error signal $e^{(n)}(k)$ can be written as

$$e^{(n)}(k) = [y(k)y(k-1)\dots y(k-n)] \begin{bmatrix} w_0^{(n)} \\ w_1^{(n)} \\ \vdots \\ w_n^{(n)} \end{bmatrix} \quad (4-4)$$

where

$$\begin{aligned} w_0^{(n)} &= -k^{(n)} = 1 \\ w_1^{(n)} &= \sum_{i=1}^n k^{(i)} k^{(i-1)} = \sum_{i=0}^{n-1} k^{(i)} k^{(i+1)} \\ &\vdots \\ w_n^{(n)} &= -k^{(n)} \end{aligned} \quad (4-5)$$

$w_i^{(n)}$ is the summation of the gain factors of all transmission paths with delay i from the input to $e^{(n)}(k)$. A recursive formulation for $w_i^{(n)}$ is now developed. Equation (4-4) can be written

$$e^{(n)}(k) = \underline{y}^{(n)T}(k) \underline{w}^{(n)} \quad (4-6)$$

where

$$\underline{y}^{(n)T}(k) = [y(k)y(k-1)\dots y(k-n)] \quad (4-7)$$

and

$$\underline{w}^{(n)T} = [w_0^{(n)} w_1^{(n)} \dots w_n^{(n)}] \quad (4-8)$$

It should be noted that the $\underline{y}^{(n)}(k)$ and $\underline{w}^{(n)}$ are $(n+1)$ terms long.

Similarly $\bar{e}^{(n)}(k)$ can be written

$$\bar{e}^{(n)}(k) = [y(k)y(k-1)\dots y(k-n)] \begin{bmatrix} w_n^{(n)} \\ w_{n-1}^{(n)} \\ \vdots \\ w_0^{(n)} \end{bmatrix} \quad (4-9)$$

or

$$\bar{e}^{(n)}(k) = \underline{y}^{(n)T}(k) \underline{\tilde{w}}^{(n)} \quad (4-10)$$

where $\underline{\tilde{w}}^{(n)}$ is the inverted version of $\underline{w}^{(n)}$. With this notation, a delay of the backward error sequence is represented by a shift downward of the elements of the weighting vector $\underline{\tilde{w}}^{(n)}$, with a zero shifted into the top, or

$$\bar{e}^{(n)}(k-1) = \underline{y}^{(n+1)T}(k) \begin{bmatrix} 0 \\ \underline{\tilde{w}}^{(n)} \end{bmatrix} \quad (4-11)$$

The weight vector associated with $e^{(n)}(k)$ or $\bar{e}^{(n)}(k)$ can be found recursively. Substituting (4-6) and (4-11), the equations for the signals $e^{(n)}(k)$ and $\bar{e}^{(n)}(k-1)$, into the forward error recursion (4-1), yields

$$\begin{aligned} e^{(n+1)}(k) &= \underline{y}^{(n)T} \underline{w}^{(n)} - k^{(n+1)} \underline{y}^{(n+1)T} \begin{bmatrix} 0 \\ \underline{\tilde{w}}^{(n)} \end{bmatrix} \\ &= \underline{y}^{(n+1)T} \begin{bmatrix} \underline{w}^{(n)} \\ 0 \end{bmatrix} - k^{(n+1)} \begin{bmatrix} 0 \\ \underline{\tilde{w}}^{(n)} \end{bmatrix} \end{aligned} \quad (4-12)$$

Comparing (4-12) with (4-6) rewritten for order (n+1), yields

$$\underline{w}^{(n+1)} = \begin{bmatrix} \underline{w}^{(n)} \\ \hline 0 \end{bmatrix} - k^{(n+1)} \begin{bmatrix} 0 \\ \underline{\tilde{z}}(\underline{n}) \\ \underline{w} \end{bmatrix} \quad (4-13)$$

Similarly, substituting (4-6) and (4-11) into (4-2), the backward error recursion, results in

$$\begin{aligned} \underline{\tilde{e}}^{(n)}(k) &= \underline{y}^{(n+1)}(k) \begin{bmatrix} 0 \\ \underline{\tilde{z}}(\underline{n}) \\ \underline{w} \end{bmatrix} - k^{(n+1)} \underline{y}^{(n)}(k) \underline{w}^{(n)} \\ &= \underline{y}^{(n+1)}(k) \left[\begin{bmatrix} 0 \\ \underline{\tilde{z}}(\underline{n}) \\ \underline{w} \end{bmatrix} - k^{(n+1)} \begin{bmatrix} \underline{w}^{(n)} \\ \hline 0 \end{bmatrix} \right] \quad (4-14) \end{aligned}$$

Comparing this with (4-11) gives the recursion

$$\underline{\tilde{w}}^{(n+1)} = \begin{bmatrix} 0 \\ \underline{\tilde{z}}(\underline{n}) \\ \underline{w} \end{bmatrix} - k^{(n+1)} \begin{bmatrix} \underline{w}^{(n)} \\ \hline 0 \end{bmatrix} \quad (4-15)$$

It should be noted that the initial conditions for the weight vectors $\underline{w}^{(0)}$ and $\tilde{\underline{w}}^{(0)}$ are

$$\underline{w}^{(0)} = [1] \quad \tilde{\underline{w}}^{(0)} = [1] \quad (4-16)$$

As an illustration, from Figure 4.1 it can be seen that the second stage weight vectors can be written as

$$\underline{w}^{(2)} = \begin{bmatrix} 1 \\ -k^{(1)} + k^{(1)}k^{(2)} \\ -k^{(2)} \end{bmatrix} \quad \tilde{\underline{w}}^{(2)} = \begin{bmatrix} -k^{(2)} \\ -k^{(1)} + k^{(1)}k^{(2)} \\ 1 \end{bmatrix} \quad (4-17)$$

Applying these to (4-13) yields

$$\begin{aligned} \underline{w}^{(3)} &= \begin{bmatrix} 1 \\ -k^{(1)} + k^{(1)}k^{(2)} \\ k^{(2)} \\ \hline 0 \end{bmatrix} - k^{(3)} \begin{bmatrix} 0 \\ \hline -k^{(2)} \\ -k^{(1)} + k^{(1)}k^{(2)} \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} 1 \\ -k^{(1)} + k^{(1)}k^{(2)} + k^{(2)}k^{(3)} \\ k^{(2)} + k^{(1)}k^{(3)} - k^{(1)}k^{(2)}k^{(3)} \\ -k^{(3)} \end{bmatrix} \end{aligned} \quad (4-18)$$

and to (3-15)

$$\begin{aligned} \underline{\tilde{w}}^{(3)} &= \begin{bmatrix} 0 \\ \hline -k^{(2)} \\ -k^{(1)} + k^{(1)}k^{(2)} \\ 1 \end{bmatrix} -k^{(3)} \begin{bmatrix} 1 \\ -k^{(1)} + k^{(1)}k^{(2)} \\ -k^{(2)} \\ \hline 0 \end{bmatrix} \\ &= \begin{bmatrix} -k^{(3)} \\ -k^{(2)} + k^{(1)}k^{(3)} - k^{(1)}k^{(2)}k^{(3)} \\ -k^{(1)} + k^{(1)}k^{(2)} + k^{(2)}k^{(3)} \\ 1 \end{bmatrix} \quad (4-20) \end{aligned}$$

yielding the third stage weight vectors.

The numerator of equation (4-3) can now be rewritten using the definition of the error signals in terms of the weight vectors

$$\begin{aligned} \epsilon \{e^{(n)}(k) \bar{e}^{(n)}(k-1)\} &= \epsilon \left\{ \left[\underline{w}^{(n)T} \mid 0 \right] \underline{y}^{(n+1)}(k) \underline{y}^{(n+1)T}(k) \begin{bmatrix} 0 \\ \underline{\tilde{w}}^{(n)} \end{bmatrix} \right\} \\ &= [w_0^{(n)} w_1^{(n)} \dots w_n^{(n)} 0] \begin{bmatrix} R_{yy}(0) \dots R_{yy}(-n-1) \\ \vdots \\ R_{yy}(n+1) \dots R_{yy}(0) \end{bmatrix} \begin{bmatrix} 0 \\ w_n^{(n)} \\ \vdots \\ w_1^{(n)} \\ w_0^{(n)} \end{bmatrix} \end{aligned}$$

$$= [\underline{w}^{(n)T} \mid 0] R_{yy} \begin{bmatrix} 0 \\ -\underline{\bar{w}}^{(n)} \end{bmatrix} \quad (4-20)$$

It is seen that the numerator of (4-3), required for calculation of $k^{(n)}$, can be calculated with no presumption of symmetry or requirement that the correlation matrix be Toeplitz. The lattice method may still be used if the signals under analysis are not stationary or estimates of the correlation matrix are available that do not result in a Toeplitz and symmetric matrix. However, if the correlation matrix of (4-20) is Toeplitz and symmetric the matrix multiplication may be written as summations.

$$\begin{aligned} \varepsilon\{e^{(n)}(k)\bar{e}^{(n)}(k-1)\} &= R_{yy}(0) \sum_{i=1}^n w_i w_{n-i+1} \\ &+ R_{yy}(1) \left[\sum_{i=0}^n w_i w_{n-i} + \sum_{i=2}^n w_i w_{n-i+2} \right] \\ &+ R_{yy}(2) \left[\sum_{i=0}^{n-1} w_i w_{n-i-1} + \sum_{i=3}^n w_i w_{n-i-3} \right] \\ &+ \dots + R_{yy}(n+1) w_0^2 \end{aligned}$$

These summations are recognized as convolutions of the weight vector $\underline{w}^{(n)}$, or the correlation of the forward and backward weight vectors.

Defining correlation functions of $\underline{w}^{(n)}$ and $\tilde{\underline{w}}^{(n)}$ as

$$\begin{aligned} \phi^{(n)}(\ell) &= \sum_{i=0}^{n+1-\ell} w_i w_{n+1-\ell-i}, \quad \ell=1,2,\dots,n+1 \\ &= \begin{cases} \sum_{i=0}^{\frac{n-\ell}{2}} w_i w_{n+1-\ell-i}, & \ell=1,2,\dots,n+1 \\ & \text{and } (n-\ell) \text{ even} \\ \sum_{i=0}^{\frac{n+1-\ell}{2}} w_i w_{n+1-\ell-i} + \frac{w_{\frac{n+1-\ell}{2}}^2}{2}, & \ell=1,2,\dots,n+1 \\ & \text{and } (n-\ell) \text{ odd} \end{cases} \end{aligned} \quad (4-21)$$

$$\begin{aligned} \phi^{(n)}(\ell) &= \sum_{i=1-\ell}^n w_i w_{n+1-\ell-i}, \quad \ell=n,n+1,\dots,-1,0 \\ &= \begin{cases} \sum_{i=1-\ell}^{\frac{n-n+\ell}{2}} w_i w_{n+1-\ell-i}, & \ell=n,n+1,\dots,-1,0 \\ & \text{and } (n+\ell) \text{ even} \\ \sum_{i=1-\ell}^{\frac{n-n+\ell+1}{2}} w_i w_{n+1-\ell-i} + \frac{w_{\frac{n-n+\ell+1}{2}}^2}{2}, & \ell=n,n+1,\dots,-1,0 \\ & \text{and } (n+\ell) \text{ odd} \end{cases} \end{aligned} \quad (4-22)$$

$$\phi^{(n)}(-n-1) = 0 \quad (4-23)$$

Equation (4-20) can now be rewritten as

$$\begin{aligned} \epsilon \{ e^{(n)}(k) \bar{e}^{(n)}(k-1) \} = \\ R_{yy}(0) \phi^{(n)}(0) + R_{yy}(1) [\phi^{(n)}(1) + \phi^{(n)}(-1)] \\ + R_{yy}(2) [\phi^{(n)}(2) + \phi^{(n)}(-2)] \\ + \dots + \\ + R_{yy}(n+1) [\phi^{(n)}(n+1) + \phi^{(n)}(-n-1)] \quad (4-24) \end{aligned}$$

This expression can be written more compactly as the product of two vectors, one with elements of R_{yy} , the other of ϕ .

$$\begin{aligned} \epsilon \{ e^{(n)}(k) \bar{e}^{(n)}(k-1) \} = \\ [R_{yy}(0) \dots R_{yy}(n+1)] \begin{bmatrix} \phi^{(n)}(0) \\ \phi^{(n)}(1) + \phi^{(n)}(-1) \\ \vdots \\ \phi^{(n)}(n+1) + \phi^{(n)}(-n-1) \end{bmatrix} \quad (4-25) \end{aligned}$$

or

$$\varepsilon\{e^{(n)}(k)\bar{e}^{(n)}(k-1)\} = \underline{r}^{(n)T} \underline{w}^{(n)} \quad (4-26)$$

where

$$\underline{r}^{(n)T} = [R_{yy}(0) \dots R_{yy}(n+1)] \quad (4-27)$$

and

$$\underline{w}^{(n)} = \begin{bmatrix} \phi^{(n)}(0) \\ \phi^{(n)}(1) + \phi^{(n)}(-1) \\ \vdots \\ \phi^{(n)}(n+1) + \phi^{(n)}(-n-1) \end{bmatrix} \quad (4-28)$$

The denominator of equation (4-3) could be found using similar arguments, but generally offers no advantage to previously described methods. The recursion

$$\varepsilon\{e^{(0)}(k)e^{(0)}(k)\} = R_{yy}(0) = P^{(0)}$$

$$\varepsilon\{e^{(n)}(k)e^{(n)}(k)\} = P^{(n-1)}(1-k^{(n)2}) = P^{(n)} \quad (4-29)$$

should be used, Equation (4-3) may now be rewritten using (4-20) or (4-26) and (4-29).

$$k^{(n+1)} = \left\{ \frac{\underline{r}^{(n)T} \underline{w}^{(n)}}{p^{(n)}} \right. \quad \begin{array}{l} \text{For } \underline{R} \text{ Toeplitz and} \\ \text{symmetric} \end{array} \quad (4-30a)$$

$$\frac{\left[\underline{w}^{(n)T} \mid 0 \right] \underline{R} \begin{bmatrix} 0 \\ \underline{w}^{(n)} \end{bmatrix}}{p^{(n)}} \quad \begin{array}{l} \text{For general} \\ \underline{R} \end{array} \quad (4-30b)$$

1. Implementation of the Single Channel Algorithm

The lattice can be implemented using correlation weights by applying (4-16), (4-13), (4-27), (4-20) or (4-28), and (4-29) and (4-30). A flowchart of the procedure is presented as Figure 4.2.

For an mth order analysis the procedure will be:

- (1) For stationary or windowed signals find the correlation vector $\underline{r}^{(n)}$. For nonstationary systems or infinite unwindowed data find the correlation matrix $\underline{R}^{(n)}$.
- (2) Find $\underline{w}^{(n)}$. For the first stage of analysis the initial conditions of (4-16) is used. For later stages (4-13) gives the recursive solution for $\underline{w}^{(n)}$.
- (3) The numerator of (4-30) is found using the weight vector and either a correlation vector $\underline{r}^{(n)}$ or the correlation matrix $\underline{R}^{(n)}$.

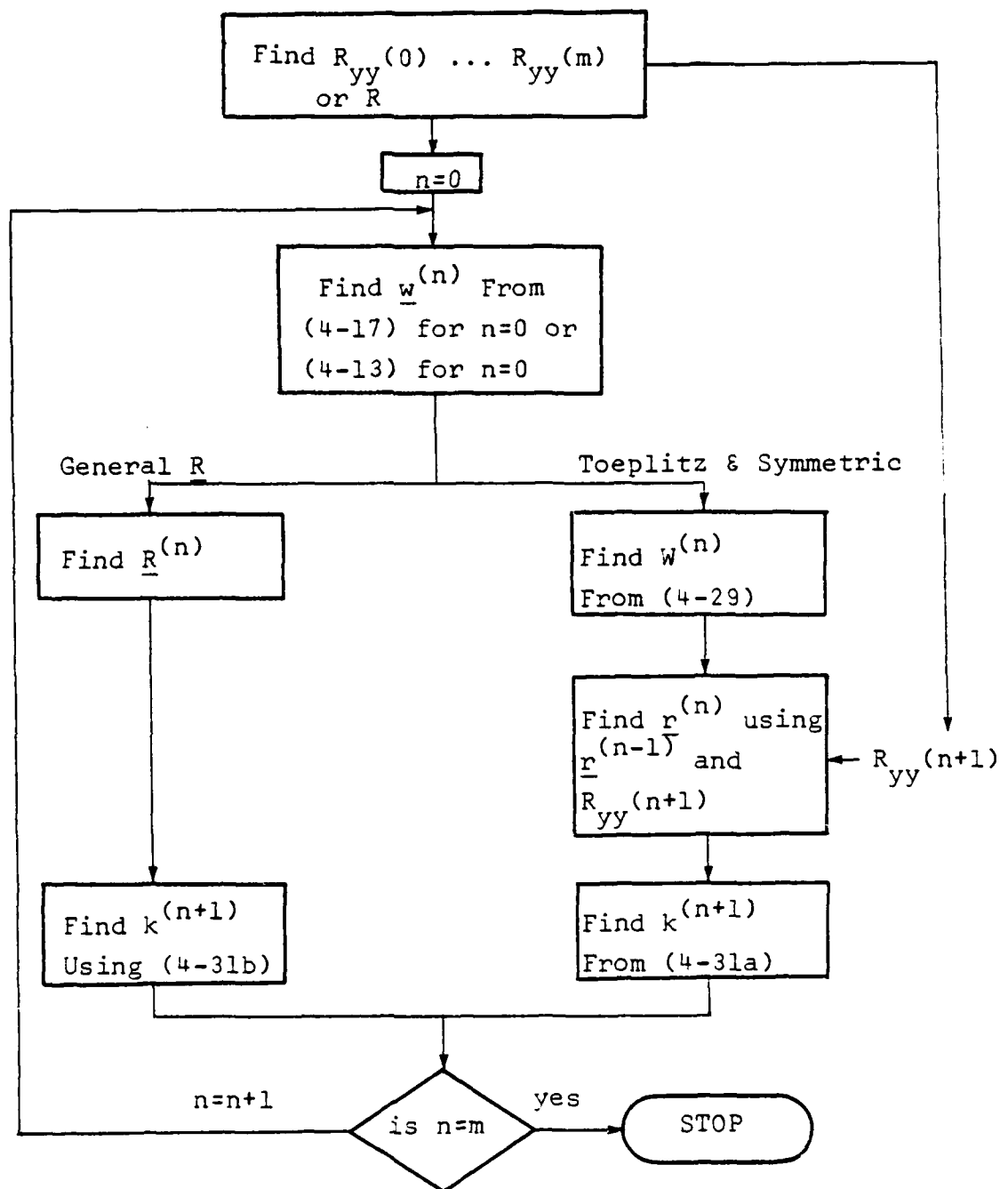


Figure 4.2 Flow Chart for mth Order Analysis

- (4) Find $\underline{p}^{(n)}$ from (4-28). For $n=0$, $\underline{p}^{(0)} = R_{yy}(0)$;
otherwise use the recursion of (4-28).
- (5) $k^{(n+1)}$ is now found from (4-30).
- (6) If higher order analysis is desired, increment
n and return to (2).

Figure 4.3 gives a comparison of the number of multiplications required to calculate reflection factors for a seven stage lattice that uses 256 data points, and assumes a stationary system with the data windowed to ensure a Toeplitz structure.

CONDITIONS: 7 stages
256 data points
Data windowed for Toeplitz and symmetric
correlation matrix

TIME DOMAIN LATTICE:

$$\begin{aligned} \# \text{ multiplications} &= l + (l+2)n + l(n-1)2 = \\ &\quad \begin{matrix} p^{(0)} & k^{(n)} & \text{error signal} \\ & & \text{update} \end{matrix} \\ &= 256 + (256+2)7 + 256(6)2 = 5134 \end{aligned}$$

For Correlation Lattice

$$\# \text{ multiplications} = l + \sum_{n=0}^{m-1} (n+1) + \sum_{n=1}^m \frac{(n+1)^2}{2} + ml + \sum_{n=1}^m (n+1)$$

$$\begin{aligned} R_{yy}(0) \quad \underline{w}^{(n)} &\approx \underline{w}^{(n)} \quad R_{yy}(n) k^{(n)} \\ &= 256 + 28 + 140 + 1792 + 34 = 2250 \end{aligned}$$

$$\frac{2250}{5134} \approx 44\%$$

Figure 4.3 Multiplications Required for Implementation

B. THE TWO CHANNEL LATTICE

The two channel lattice developed in II,C can similarly be redefined in terms of correlation weights, the details of which are presented in Appendix A. The following is a summary of the steps and equations necessary to implement the algorithm.

To implement the two channel algorithm it is necessary to calculate eight weighting vectors, two for each of the four forward and backward error signals. These can be found using the initial conditions

$$\underline{w}_{yy}^{(0)} = \underline{w}_{uu}^{(0)} = [1], \quad \underline{w}_{yu}^{(0)} = \underline{w}_{uy}^{(0)} = [0] \quad (4-31)$$

$$\bar{\underline{w}}_{yy}^{(0)} = \bar{\underline{w}}_{uu}^{(0)} = [1], \quad \bar{\underline{w}}_{yu}^{(0)} = \bar{\underline{w}}_{uy}^{(0)} = [0] \quad (4-32)$$

and the recursions for higher orders

$$\begin{bmatrix} \underline{w}_{yy}^{(n+1)} \\ \underline{w}_{yu}^{(n+1)} \\ \underline{w}_{uy}^{(n+1)} \\ \underline{w}_{uu}^{(n+1)} \end{bmatrix} = \begin{bmatrix} \underline{w}_{yy}^{(n)} \\ \underline{w}_{yu}^{(n)} \\ \underline{w}_{uy}^{(n)} \\ \underline{w}_{uu}^{(n)} \end{bmatrix} - \underline{K}^{(n+1)T} \begin{bmatrix} 0 \\ \bar{\underline{w}}_{yy}^{(n)} \\ 0 \\ \bar{\underline{w}}_{yu}^{(n)} \\ 0 \\ \bar{\underline{w}}_{uy}^{(n)} \\ 0 \\ \bar{\underline{w}}_{uu}^{(n)} \end{bmatrix} \quad (4-33)$$

and

$$\begin{bmatrix} \bar{w}_{yy}^{(n+1)} \\ \bar{w}_{yu}^{(n+1)} \\ \bar{w}_{uy}^{(n+1)} \\ \bar{w}_{uu}^{(n+1)} \end{bmatrix} = \begin{bmatrix} 0 \\ \bar{w}_{yy}^{(n)} \\ 0 \\ \bar{w}_{yu}^{(n)} \\ 0 \\ \bar{w}_{uy}^{(n)} \\ 0 \\ \bar{w}_{uu}^{(n)} \end{bmatrix} - \underline{K}^{(n+1)T} \begin{bmatrix} w_{yy}^{(n)} \\ 0 \\ w_{yu}^{(n)} \\ 0 \\ w_{uy}^{(n)} \\ 0 \\ w_{uu}^{(n)} \\ 0 \end{bmatrix} \quad (4-34)$$

The vectors of (4-33) and (4-34) each have two elements which are also vectors, and the matrix multiplication is executed as a (2x2) matrix times a (2x1) vector.

The (2x2) matrix $\underline{\Delta}^{(n)}$ can be found using these weight vectors and a correlation matrix $\underline{R}^{(n)}$.

$$\underline{R}^{(n)} = \begin{bmatrix} R_{yy}^{(n)} & R_{yu}^{(n)} \\ R_{uy}^{(n)} & R_{uu}^{(n)} \end{bmatrix} \quad (4-35)$$

where

$$\underline{R}_{xz}^{(n)} = \begin{pmatrix} R_{xz}^{(0)} & \dots & R_{xz}^{(-n-1)} \\ \vdots & & \vdots \\ R_{xz}^{(n+1)} & \dots & R_{xz}^{(0)} \end{pmatrix} \quad (4-36)$$

The four elements of the $\underline{\Delta}^{(n)}$ matrix are defined by the relations

$$\Delta_{11} = [\underline{w}_{yy}^{(n)T} \mid \underline{w}_{yu}^{(n)T} \mid 0] \underline{R}^{(n)} \begin{bmatrix} 0 \\ \underline{w}_{yy}^{(n)} \\ \underline{w}_{yu}^{(n)} \\ 0 \end{bmatrix} \quad (4-37)$$

$$\Delta_{12} = [\underline{w}_{yy}^{(n)T} \mid \underline{w}_{yu}^{(n)T} \mid 0] \underline{R}^{(n)} \begin{bmatrix} 0 \\ \underline{w}_{uy}^{(n)} \\ \underline{w}_{uu}^{(n)} \\ 0 \end{bmatrix} \quad (4-38)$$

$$\Delta_{21} = [\underline{w}_{uy}^{(n)T} \mid \underline{w}_{uu}^{(n)T} \mid 0] \underline{R}^{(n)} \begin{bmatrix} 0 \\ \underline{w}_{yy}^{(n)} \\ \underline{w}_{yu}^{(n)} \\ 0 \end{bmatrix} \quad (4-39)$$

$$\Delta_{22} = [\underline{w}_{uy}^{(n)T} \mid \underline{w}_{uu}^{(n)T} \mid 0] \underline{R}^{(n)} \begin{bmatrix} 0 \\ \underline{w}_{uy}^{(n)} \\ \underline{w}_{uu}^{(n)} \\ 0 \end{bmatrix} \quad (4-40)$$

The $\underline{P}^{(n)}$ and $\underline{\bar{P}}^{(n)}$ matrices are found using the recursions previously cited, namely

$$\underline{P}^{(0)} = \underline{\bar{P}}^{(0)} = \begin{bmatrix} R_{yy}^{(0)} & R_{yu}^{(0)} \\ R_{uy}^{(0)} & R_{uu}^{(0)} \end{bmatrix} \quad (4-41)$$

$$\underline{P}^{(n+1)} = \underline{P}^{(n)} [I - \underline{\bar{K}}^{(n)} \underline{K}^{(n)}] \quad (4-42)$$

$$\underline{\bar{P}}^{(n+1)} = \underline{\bar{P}}^{(n)} [I - \underline{K}^{(n)} \underline{\bar{K}}^{(n)}] \quad (4-43)$$

The lattice parameters are found using the $\underline{\Delta}^{(n)}$, $\underline{P}^{(n)}$ and $\underline{\bar{P}}^{(n)}$ matrices from

$$\underline{K}^{(n+1)} = \underline{\bar{P}}^{(n)-1} \underline{\Delta}^{(n)T} \quad (4-44)$$

$$\underline{\bar{K}}^{(n+1)} = \underline{P}^{(n)-1} \underline{\Delta}^{(n)} \quad (4-45)$$

A flow chart of this implementation is presented as Figure 4.4.

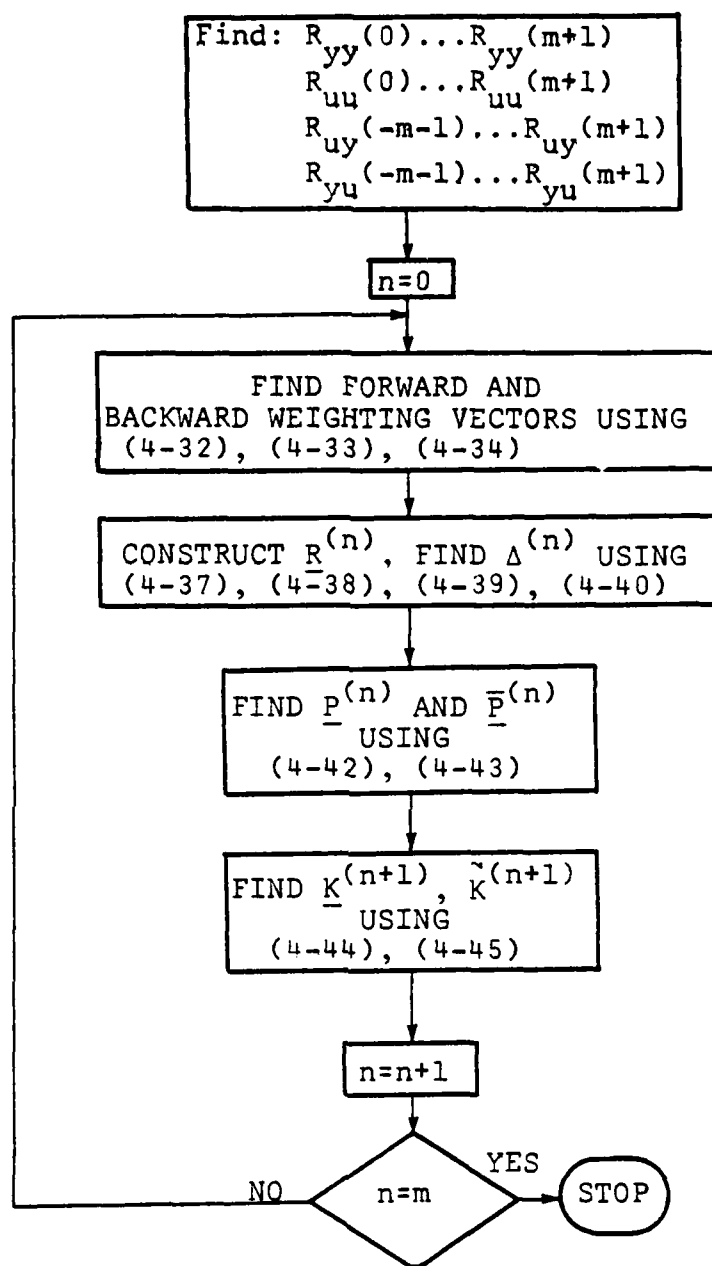


Figure 4.4 The m-order Two Channel Implementation

1. Correlation Lattice Simulations

To verify the algorithms presented in the forgoing a FORTRAN program was written. This program implements the flowchart of Figure 4.4. The correlation matrices generated by this program are both Toeplitz and symmetric. The examples here are the same two systems used in Chapter III.E. and represent a run which is typical of the ensemble of tests that were made. The distribution over the ensemble was minor since measurement noise was not considered. The fourth order model of System 1 is shown with Gaussian elimination using 256 points, in Figure 4.5. Figure 4.5 shows the results using 4096 points. Figure 4.5 shows the same comparison of System 2 using 128 data points, and Figure 4.8 is System 2 modeled with 1024 points. In almost all cases the results of the correlation method are essentially identical to the results of the Gaussian elimination method. This should not be unexpected. Both matrix manipulation and correlation lattice methods use identical data, the auto and cross correlations of the input data signals. Both methods window the data identically. Both methods are solving the same normal equations, though through differing algorithms.

The number of multiplications saved by the correlation lattice, when compared to the conventional lattice implementation, is approximately $8p$, where p is the number

of data points. This is because the error signals are not multiplied by the lattice parameter matrices $\underline{K}^{(n)}$ and $\underline{\bar{K}}^{(n)}$ in the correlation implementation.

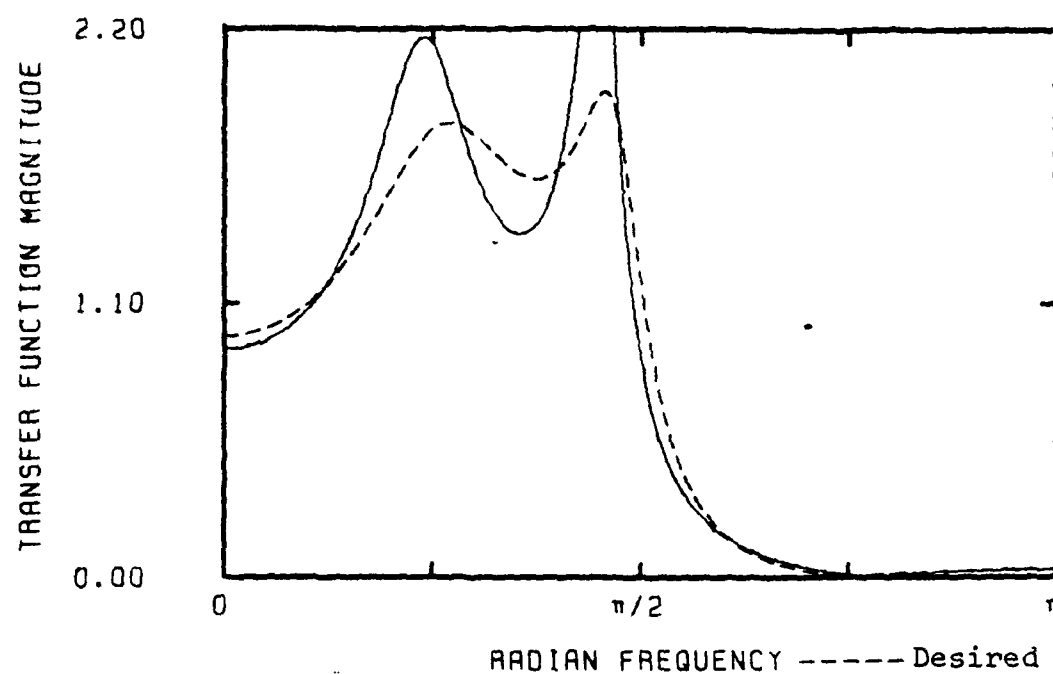


Figure 4.5a System 1, 256 Points Gaussian Elimination

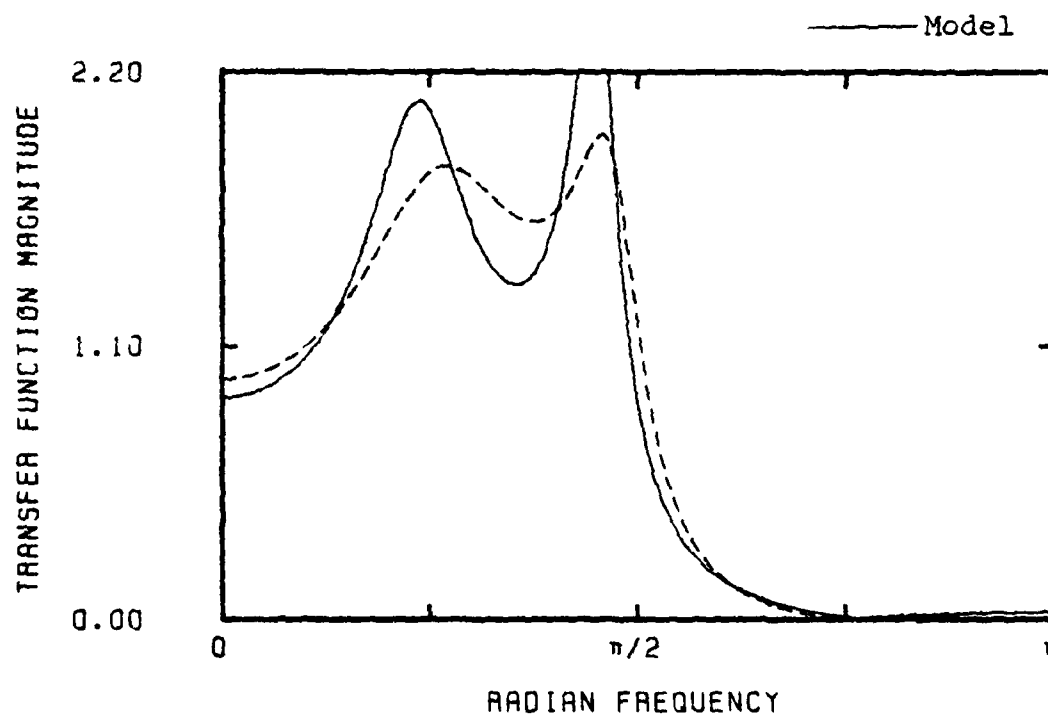


Figure 4.5b System 1, 256 Points Correlation Lattice

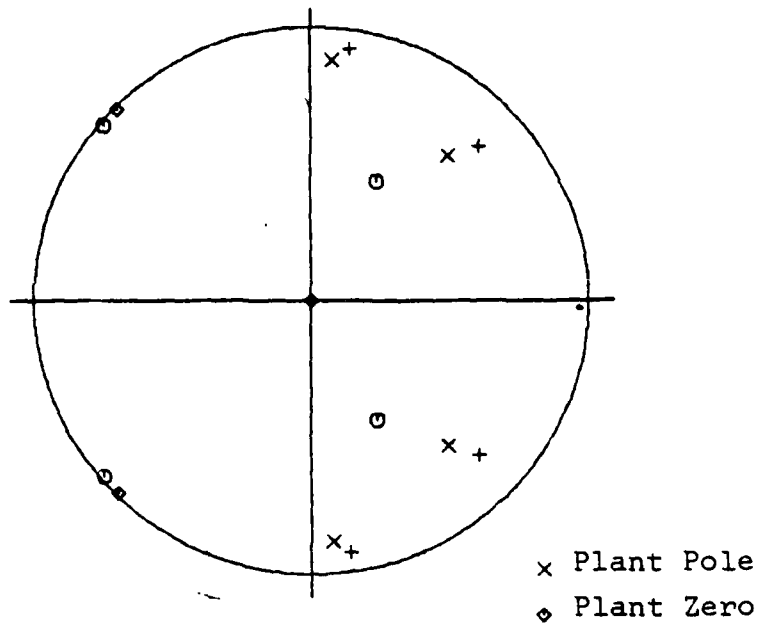


Figure 4.5c System 1, 256 Points, Gaussian Elimination

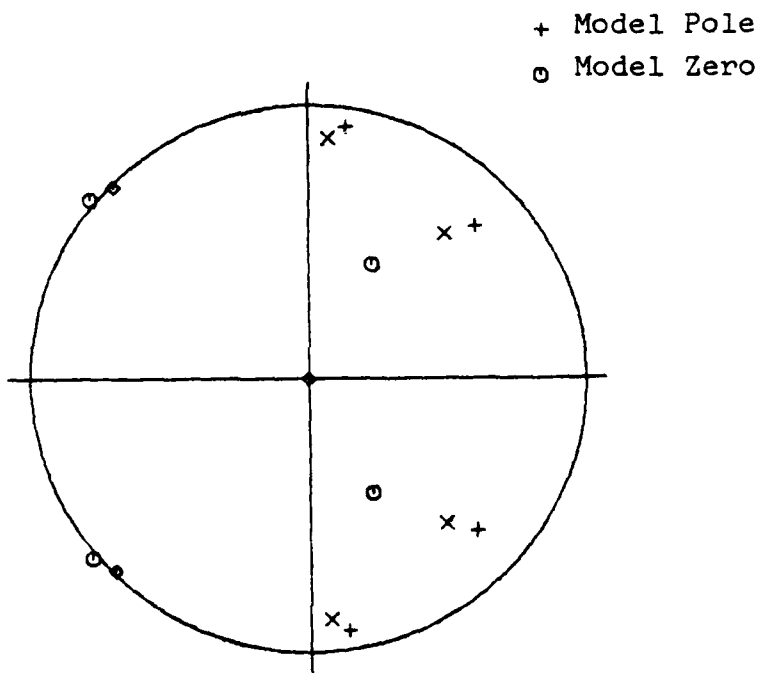


Figure 4.5d System 1, 256 Points, Correlation Lattice

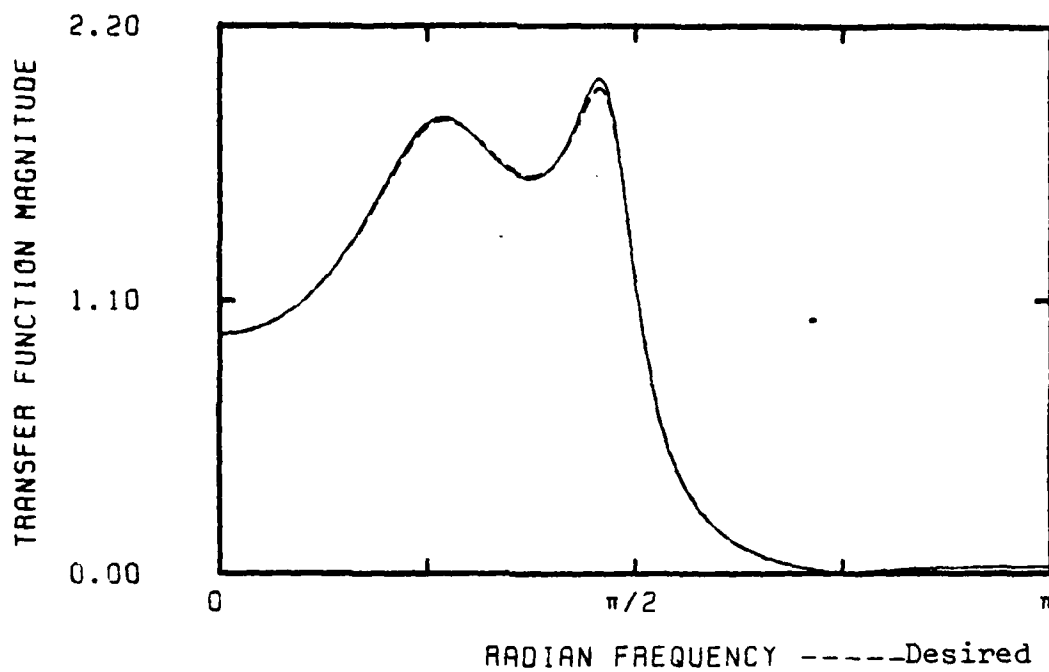


Figure 4.6a System 1, 4096 Points, Gaussian Elimination

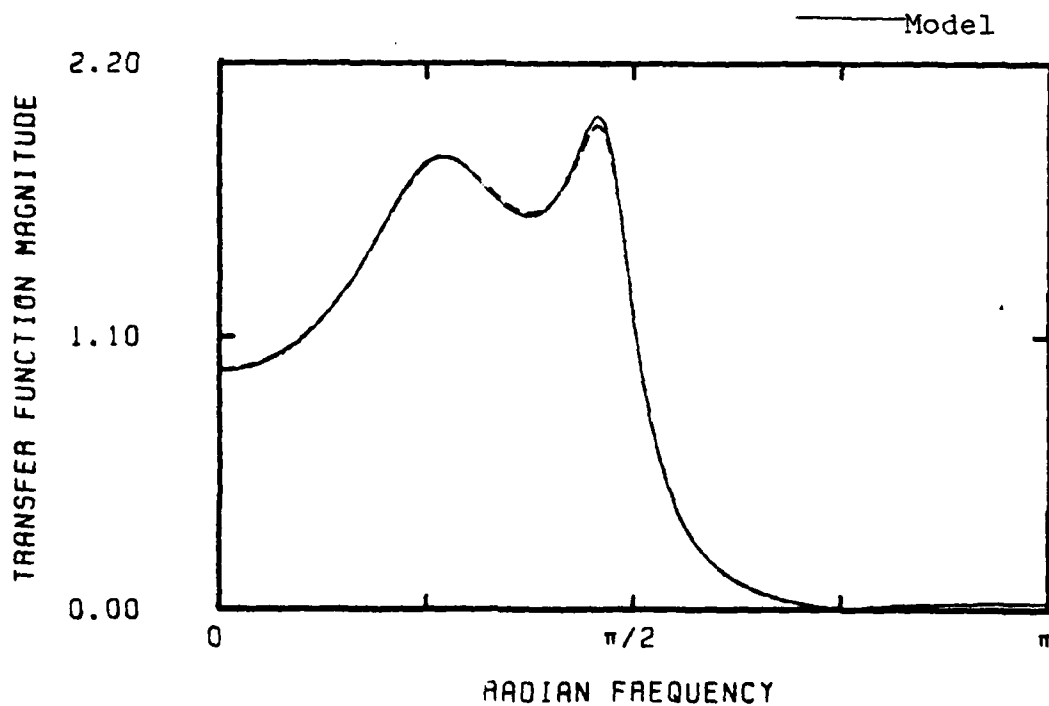
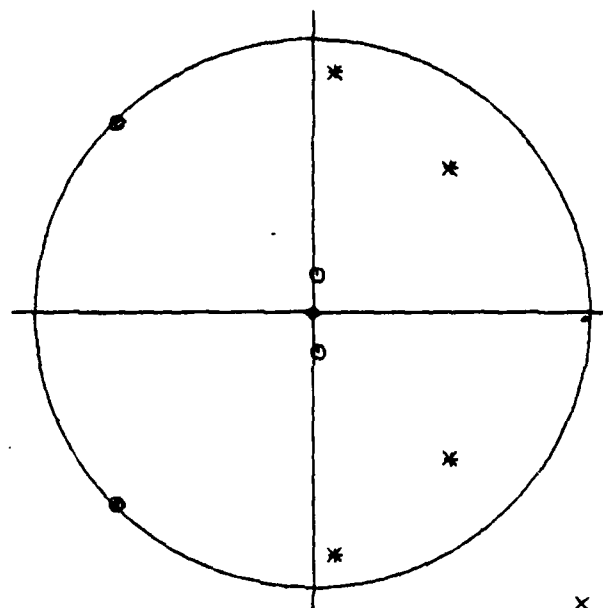


Figure 4.6b System 1, 4096 Points, Correlation Method



x Plant Pole

o Plant Zero

Figure 4.6c System 1, 4096 Points, Gaussian Elimination

+ Model Pole

o Model Zero

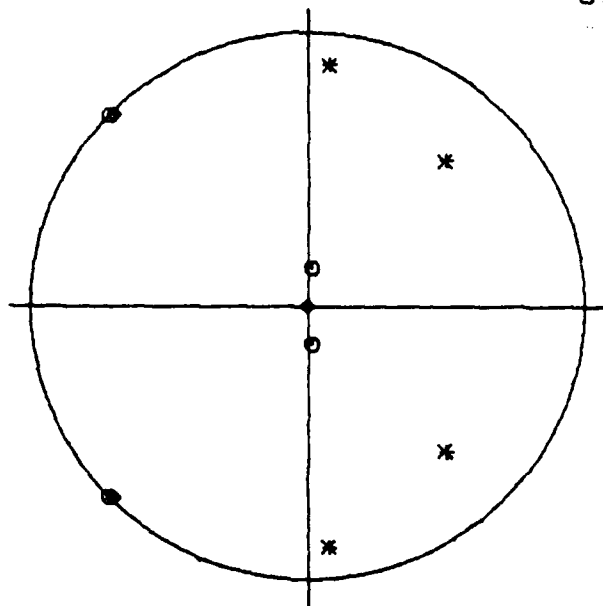


Figure 4.6d System 1, 4096 Points Correlation Lattice

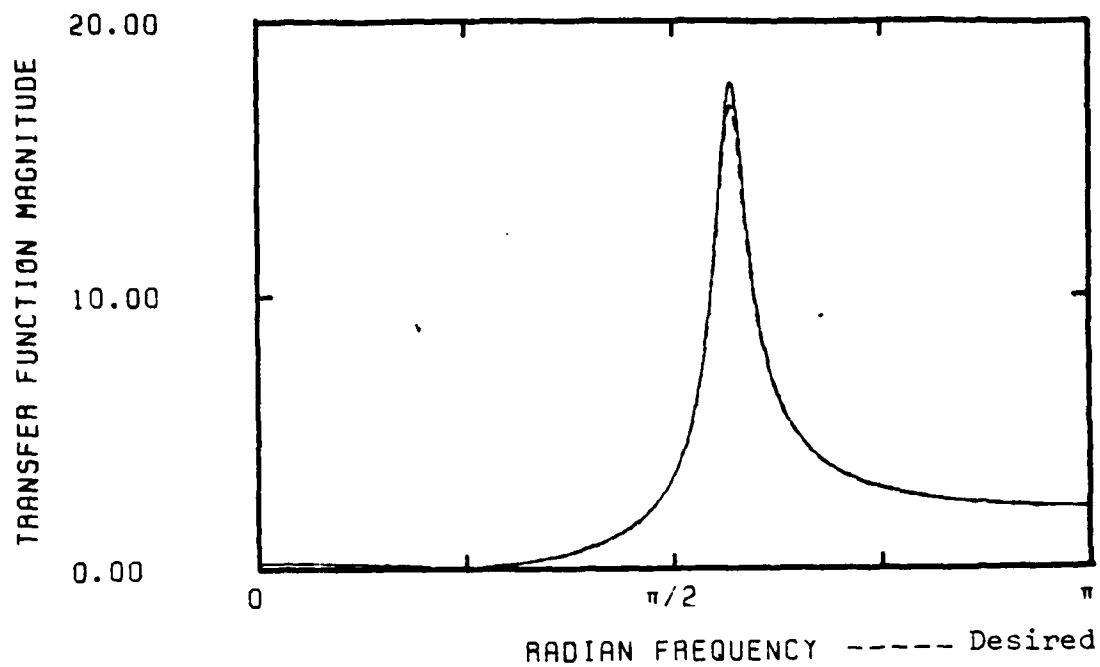


Figure 4.7a System 2, 128 Points, Gaussian Elimination

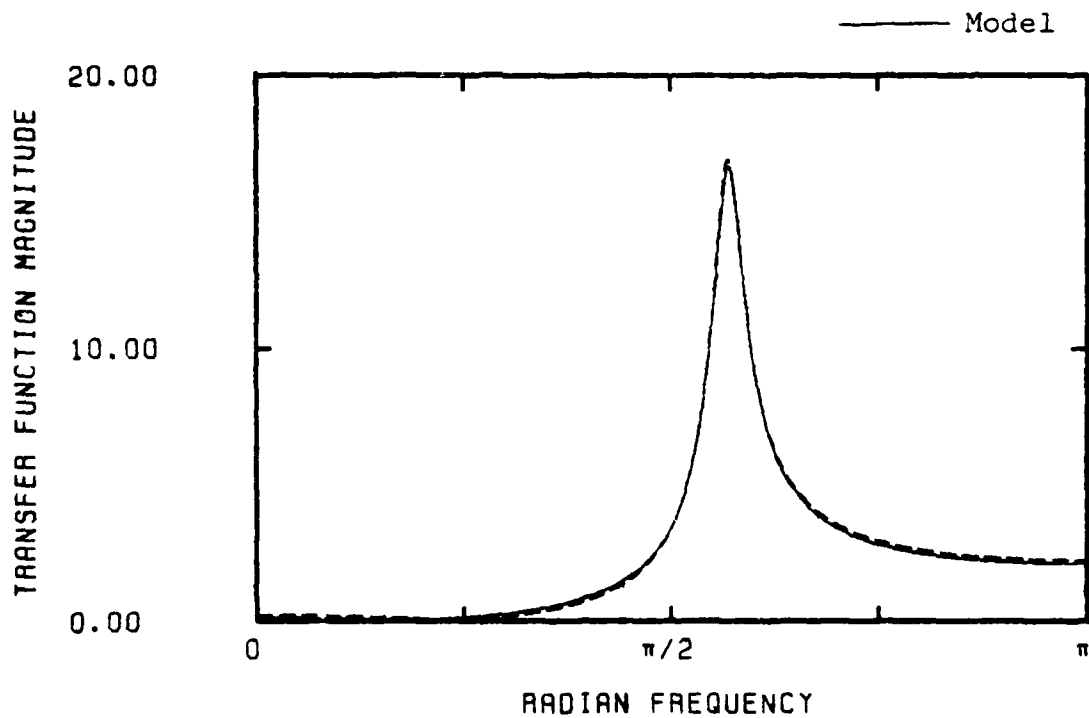
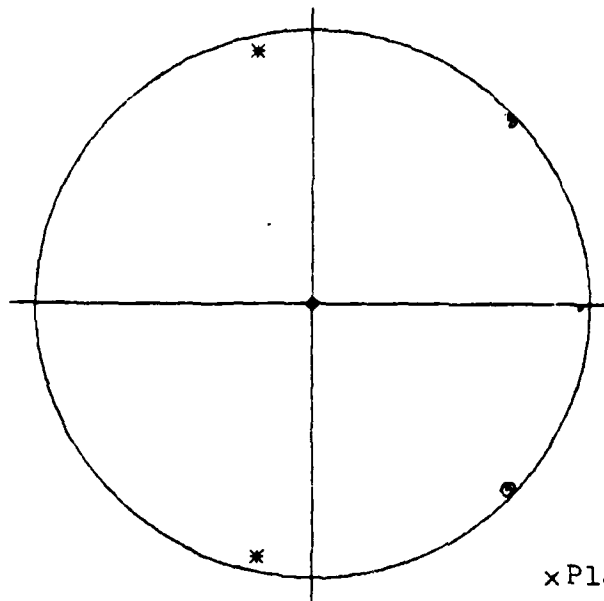


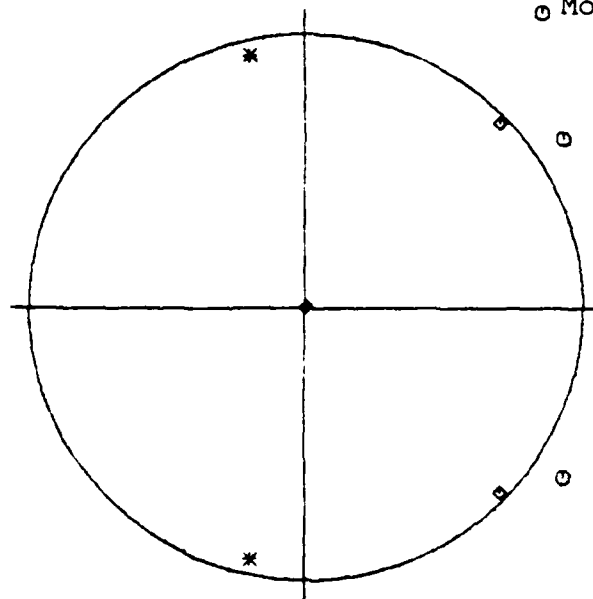
Figure 4.7b System 2, 128 Points, Correlation Method



x Plant Pole

o Plant Zero

Figure 4.7c System 2, 128 Points, Gaussian Elimination



+ Model Pole

o Model Zero

Figure 4.7d System 2, 128 Points, Batch Method

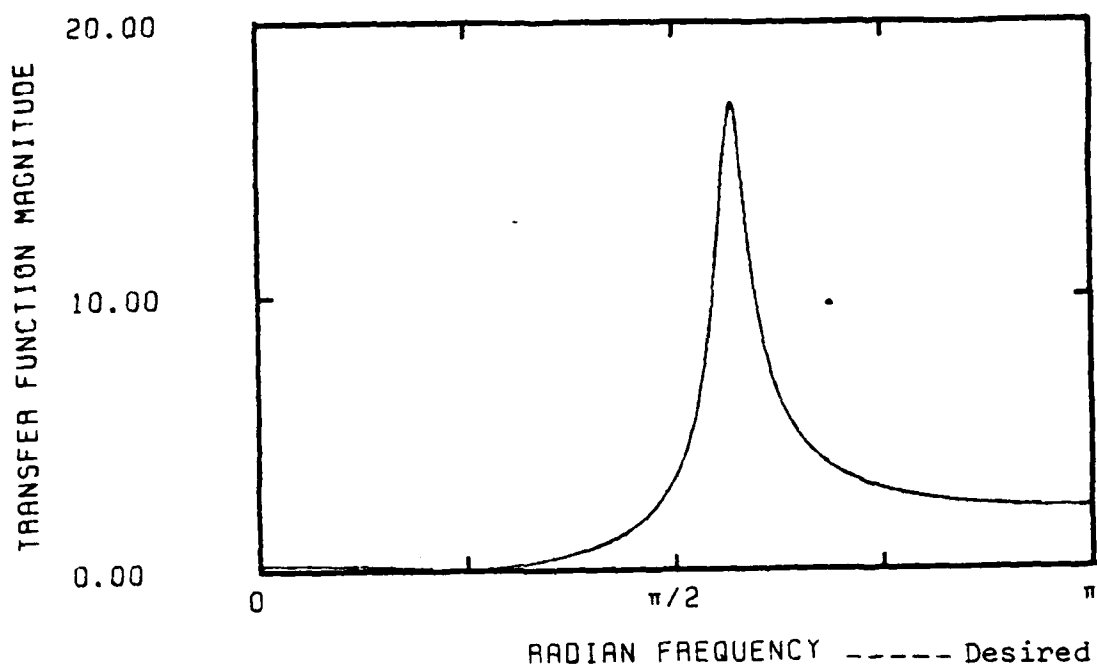


Figure 4.8a System 2, 1024 Points, Gaussian Elimination

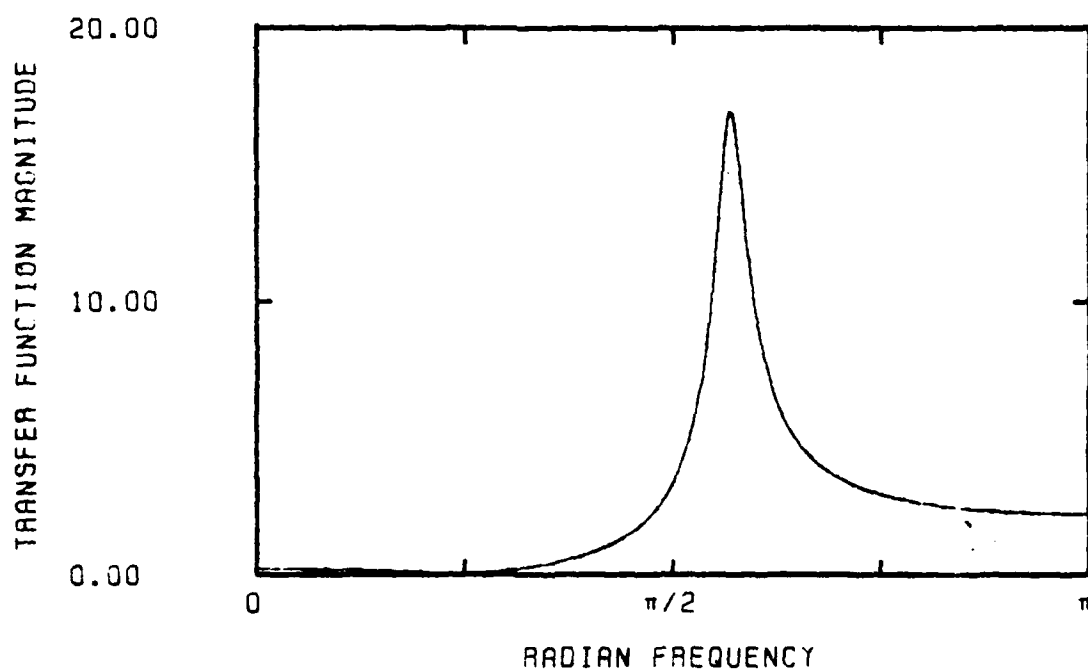
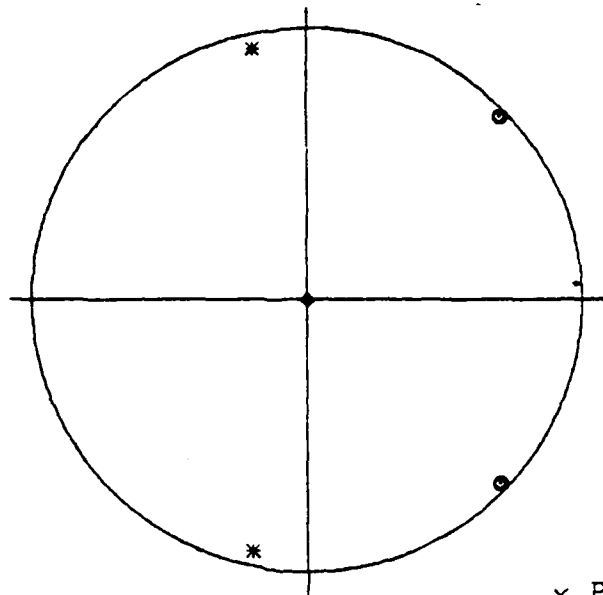


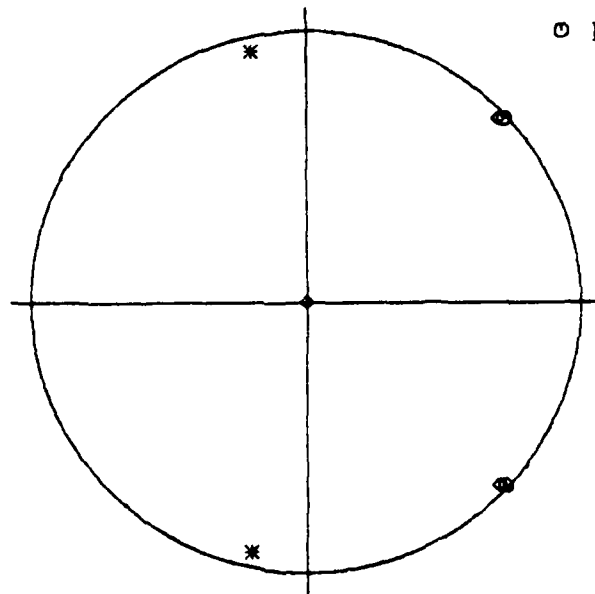
Figure 4.8b System 2, 1024 Points, Correlation Lattice



x Plant Pole

o Plant Zero

Figure 4.8c System 2, 1024 Points, Gaussian Elimination



+ Model Pole

o Model Zero

Figure 4.8d System 2, 1024 Points, Correlation Lattice

V. SUMMARY, CONCLUSIONS AND OPEN QUESTIONS

The objective of this dissertation has been to develop efficient methods of implementing the AR lattice. Two particular applications for system identification and modeling have provided the motivation for this research. The first is system fault identification. Computationally efficient algorithms must be implemented if the fault analysis is expected to be done on a real time system. Adaptive lattice structures require only moderate computation for each data point and have often been implemented in real time. The block structures, such as the conventional time domain or frequency domain lattices, require greater numbers of calculations in bursts for each block of data. These structures could be implemented using both parallel and pipeline procedures: parallel calculation of the correlations of each channel and pipelined calculation of the stages of the model.

The correlation lattice structure also encourages parallel processing implementations. In this structure calculation of the correlation matrix requires the greatest computational effort. These correlation estimates can be performed in parallel as the data becomes available.

The second application considered is the modeling of nonlinear systems. For this application the foregoing

issues are of even greater importance. Hybrid signals, products and cross products of the nonlinear system input and output, are generated and the resultant signals applied as the input of a multichannel lattice. In this case the computational and storage savings of the correlation lattice become of major significance and allow the implementation of more complex models. It has been found that the two channel lattice when applied to ARMA modeling can be simplified because of the relationships that exist between the input and the output of the system under test. The lattice can be simplified further when the system driving signal is white.

Some of the constraints found are given physical interpretations that relate to the system under analysis. There has been no such interpretation for the relationship of the lattice parameters by a factor of the DC gain of the system being modeled. It is felt that this simple relationship should have some physical interpretation related to the system under test.

The lattice was redefined using the frequency domain representation of the input data. This was found to be useful because operations in the frequency domain allowed for normalization of the input signals to whiten their spectra. This whitened spectra provides a standard input that allows lattice parameters to be used for identification

and ensures the previously mentioned relationships of the lattice parameters.

There are two normalization procedures proposed. The first requires the averaging of ensembles of the inputs to obtain an expectation of the energy spectrum of the input. The second method uses the spectrum of a single sample spectrum as a normalization function. Either of these methods will fail if the energy at any single spectral harmonic is zero.

It is not understood how either of these methods affects the accuracy of the resultant model. Experiments comparing normalization methods, with and without spectral smoothing introduced by windowing, have been inconclusive. While some examples have shown normalization can improve model accuracy over an unnormalized analysis, other examples show the opposite. It is postulated that the dynamic range of the input spectrum and normalization spectrum are of major significance, but there has been no analysis of the procedures involved.

It is also found that this frequency domain lattice can be used to synthesize a digital filter. The desired frequency domain spectrum is provided as the input of the lattice, and recursive in order lattice algorithm provides an LMS lattice model which meets the desired specification.

Lastly it is found that the lattice can be reformulated in terms of correlation weights of the input signals. This is efficient because it becomes unnecessary to generate error signals within the lattice. This results in both computational and storage efficiencies when compared to conventional lattice implementations. Though developed for the two channel lattice, the procedure can be extended to higher dimensioned lattices where the efficiencies will be of even greater significance.

Inherent in the derivation of Levinson's algorithm is the requirement that the input signals correlation matrix be symmetric Toeplitz, requiring that the signals be stationary and correlations estimates are from windowed data. When the lattice structure is derived from Levinson's algorithm the requirement of windowing the data is displaced by the lattice structure, which forces upon the data signals a different but nevertheless present window function. The symmetric Toeplitz nature of the correlation matrix is not an issue with the lattice structure.

When the lattice is reformulated in terms of correlation weights, it is again necessary to generate a correlation matrix. It does not appear in the derivation that there is a requirement for a symmetric Toeplitz matrix but it is also not understood how a nonsymmetric Toeplitz correlation matrix will affect the accuracy of the solution. This

formulation can be expected to generate errors in the solution, and it is not clear how they will propagate through the correlation lattice algorithm. This is an issue that deserve future attention.

APPENDIX A

The two channel lattice structure of Figure 2.8 can be reconfigured for a more efficient implementation using correlation weights rather than error signals. The lattice equations earlier defined in II.C are repeated here for convenience.

$$\underline{e}^{(0)}(k) = \begin{bmatrix} e_y^{(0)}(k) \\ e_u^{(0)}(k) \end{bmatrix} = \begin{bmatrix} y(k) \\ u(k) \end{bmatrix} \quad (\text{A-1})$$

$$\bar{\underline{e}}^{(0)}(k) = \begin{bmatrix} \bar{e}^{(0)}(k) \\ \bar{\underline{e}}^{(0)}(k) \end{bmatrix} = \begin{bmatrix} y(k) \\ u(k) \end{bmatrix} \quad (\text{A-2})$$

$$\underline{e}^{(n+1)}(k) = \underline{e}^{(n)}(k) - \underline{K}^{(n+1)T} \bar{\underline{e}}^{(n)}(k-1) \quad (\text{A-3})$$

$$\bar{\underline{e}}^{(n+1)}(k) = \bar{\underline{e}}^{(n)}(k-1) - \bar{\underline{K}}^{(n+1)T} \underline{e}^{(n)}(k) \quad (\text{A-4})$$

and the reflection factor solutions

$$\underline{P}^{(0)} = \bar{\underline{P}}^{(0)} = \epsilon \{ \underline{e}^{(0)}(k) \underline{e}^{(0)T}(k) \} = \begin{bmatrix} R_{yy}(0) & R_{yu}(0) \\ R_{uy}(0) & R_{uu}(0) \end{bmatrix} \quad (\text{A-5})$$

$$\underline{P}^{(n+1)} = \underline{P}^{(n)} [I - \underline{K}^{(n)} \underline{K}^{(n)}] \quad (A-6)$$

$$\underline{P}^{(n+1)} = \underline{P}^{(n)} [I - \underline{K}^{(n)} \underline{K}^{(n)}] \quad (A-7)$$

$$\underline{\Delta}^{(n)} = \epsilon \{ \underline{e}^{(n)}(k) \underline{\bar{e}}^{(n)}(k-1) \} \quad (A-8)$$

$$\underline{K}^{(n+1)} = \underline{P}^{(n)-1} \underline{\Delta}^{(n)T} \quad (A-9)$$

$$\underline{K}^{(n+1)} = \underline{P}^{(n)-1} \underline{\Delta}^{(n)} \quad (A-10)$$

A channel error signal $e_x^{(n)}(k)$ or $\bar{e}_x^{(n)}(k)$ is a linear combination of the present value and the past n values of y and u . Using this, $\underline{e}^{(n)}(k)$ may be written

$$\underline{e}^{(n)}(k) = \begin{bmatrix} e^{(n)}(k) \\ -y \\ e_u^{(n)}(k) \end{bmatrix} =$$

$$[y(k) \dots y(k-n) | u(k) \dots u(k-n)] \begin{bmatrix} w_{yy_0}^{(n)} \\ \vdots \\ w_{yy_n}^{(n)} \\ \hline w_{yu_0}^{(n)} \\ \vdots \\ w_{yu_n}^{(n)} \\ \hline w_{uy_0}^{(n)} \\ \vdots \\ w_{uy_n}^{(n)} \\ \hline w_{uu_0}^{(n)} \\ \vdots \\ w_{uu_n}^{(n)} \end{bmatrix} \quad (A-11)$$

or

$$\underline{e}^{(n)}(k) = [\underline{y}^{(n)T}(k) | \underline{u}^{(n)T}(k)] \begin{bmatrix} w_{yy}^{(n)} \\ \hline w_{yu}^{(n)} \\ \hline w_{uy}^{(n)} \\ \hline w_{uu}^{(n)} \end{bmatrix}$$

$$= \left[\begin{array}{c|c} [\underline{y}^{(n)T}(k) | \underline{u}^{(n)T}(k)] & \begin{bmatrix} w_{yy}^{(n)} \\ \text{-----} \\ w_{yu}^{(n)} \end{bmatrix} \\ \hline [\underline{y}^{(n)T}(k) | \underline{u}^{(n)T}(k)] & \begin{bmatrix} w_{uy}^{(n)} \\ \text{-----} \\ w_{uu}^{(n)} \end{bmatrix} \end{array} \right] \quad (A-12)$$

where

$$\underline{y}^{(n)T}(k) = [y(k) \dots y(k-n)] \quad (A-13)$$

$$\underline{u}^{(n)T}(k) = [u(k) \dots u(k-n)] \quad (A-14)$$

and

$$\underline{w}_{xz}^{(n)T} = [w_{xz_0}^{(n)} \dots w_{xz_n}^{(n)}] \quad (A-15)$$

Vectors $\underline{y}^{(n)}(k)$, $\underline{u}^{(n)}(k)$ and \underline{w}_{xz} are each $(n+1)$ terms long. The result of the multiplication of (A-12) is a 2 element column vector.

Using this same notation the backward delayed error vector $\underline{e}(k-1)$ may be written

$$\underline{e}^{(n)}(k-1) = \begin{bmatrix} \underline{e}_y^{(n)}(k-1) \\ \underline{e}_u^{(n)}(k-1) \end{bmatrix}$$

$$= [y(k) \dots y(k-n+1) \mid u(k) \dots u(k-n+1)] \begin{bmatrix} 0 \\ \underline{w}_{yy_0}^{(n)} \\ \vdots \\ \underline{w}_{yy_n}^{(n)} \\ \hline 0 \\ \underline{w}_{yu_0}^{(n)} \\ \vdots \\ \underline{w}_{yu_n}^{(n)} \\ \hline 0 \\ \underline{w}_{uy_0}^{(n)} \\ \vdots \\ \underline{w}_{uy_n}^{(n)} \\ \hline 0 \\ \underline{w}_{uu_0}^{(n)} \\ \vdots \\ \underline{w}_{uu_n}^{(n)} \end{bmatrix}$$

(A-16)

or

$$\bar{\underline{e}}^{(n)}(k+1) = \left[\begin{array}{c|c} [\underline{y}^{(n+1)T}(k) | \underline{u}^{(n+1)T}(k)] & \begin{bmatrix} 0 \\ \bar{\underline{w}}_{yy}^{(n)} \\ \hline 0 \\ \bar{\underline{w}}_{yu}^{(n)} \end{bmatrix} \\ \hline [\underline{y}^{(n+1)T}(k) | \underline{u}^{(n+1)T}(k)] & \begin{bmatrix} 0 \\ \bar{\underline{w}}_{uy}^{(n)} \\ \hline 0 \\ \bar{\underline{w}}_{yu}^{(n)} \end{bmatrix} \end{array} \right] \quad (\text{A-17})$$

where

$$\bar{\underline{w}}_{xz}^{(n)T} = [\bar{\underline{w}}_{xz_0}^{(n)} \dots \bar{\underline{w}}_{xz_n}^{(n)}] \quad (\text{A-18})$$

In the single channel case the backward weighting vector was equal to the forward weighting vector inverted in place. This arose because forward and backward reflection factors are equal in the single channel lattice. In the two channel lattice $\underline{k}^{(n)}$ is not in general equal to $\bar{\underline{k}}^{(n)}$, so there is no similar relationship between the forward and backward weighting vectors.

Using this notation (A-3) may be rewritten by substituting (A-12) and (A-17), adjusted for proper order,

$$\underline{e}^{(n+1)}(k) = [\underline{y}^{(n+1)T}(k) | \underline{u}^{(n+1)T}(k)] \begin{bmatrix} \underline{w}_{yy}^{(n)} \\ 0 \\ \text{---} \\ \underline{w}_{yu}^{(n)} \\ 0 \\ \text{---} \\ \underline{w}_{uy}^{(n)} \\ 0 \\ \text{---} \\ \underline{w}_{uu}^{(n)} \\ 0 \end{bmatrix} - \underline{K}^{(n)T} \begin{bmatrix} 0 \\ \underline{w}_{yy}^{(n)} \\ \text{---} \\ 0 \\ \underline{w}_{yu}^{(n)} \\ \text{---} \\ 0 \\ \underline{w}_{uy}^{(n)} \\ \text{---} \\ 0 \\ \underline{w}_{uu}^{(n)} \end{bmatrix}$$

(A-19)

Comparing (A-19) with (A-12) and adjusting for order (n+1),

$$\begin{bmatrix} \underline{w}_{yy}^{(n+1)} \\ \text{---} \\ \underline{w}_{yu}^{(n+1)} \\ \text{---} \\ \underline{w}_{uy}^{(n+1)} \\ \text{---} \\ \underline{w}_{uu}^{(n+1)} \end{bmatrix} = \begin{bmatrix} \underline{w}_{yy}^{(n)} \\ 0 \\ \text{---} \\ \underline{w}_{yu}^{(n)} \\ 0 \\ \text{---} \\ \underline{w}_{uy}^{(n)} \\ 0 \\ \text{---} \\ \underline{w}_{uu}^{(n)} \\ 0 \end{bmatrix} - \underline{K}^{(n+1)T} \begin{bmatrix} 0 \\ \underline{w}_{yy}^{(n)} \\ \text{---} \\ 0 \\ \underline{w}_{yu}^{(n)} \\ \text{---} \\ 0 \\ \underline{w}_{uy}^{(n)} \\ \text{---} \\ 0 \\ \underline{w}_{uu}^{(n)} \end{bmatrix} \quad (A-20)$$

This provides a recursive solution for the forward weight vectors.

In illustration, $e^{(0)}(k)$ is defined by (A-1) as

$$\begin{aligned} \underline{e}^{(0)}(k) &= \begin{bmatrix} y(k) \\ \hline u(k) \end{bmatrix} = [\underline{y}(k) | \underline{u}(k)] \begin{bmatrix} \begin{bmatrix} 1 \\ \hline 0 \end{bmatrix} \\ \hline \begin{bmatrix} 0 \\ \hline 1 \end{bmatrix} \end{bmatrix} \\ &= [\underline{y}^{(0)T}(k) | \underline{u}^{(0)T}(k)] \begin{bmatrix} \begin{bmatrix} w_{yy}^{(0)} \\ \hline w_{yu}^{(0)} \end{bmatrix} \\ \hline \begin{bmatrix} w_{uy}^{(0)} \\ \hline w_{uu}^{(0)} \end{bmatrix} \end{bmatrix} \end{aligned}$$

or

$$\underline{w}_{yy}^{(0)} = \underline{w}_{uu}^{(0)} = [1], \quad \underline{w}_{yu}^{(0)} = \underline{w}_{uy}^{(0)} = [0] \quad (A-21)$$

and $\bar{e}^{(o)}(k)$ can be written, with a delay added, from (A-2) as

$$\bar{e}^{(o)}(k-1) = \begin{bmatrix} y(k-1) \\ \hline u(k-1) \end{bmatrix} = [y(k)y(k-1) | u(k)u(k-1)] \begin{bmatrix} 0 \\ 1 \\ \hline 0 \\ 0 \\ \hline 0 \\ 0 \\ \hline 0 \\ 1 \end{bmatrix}$$

$$= [\underline{y}^{(1)T}(k) | \underline{u}^{(1)T}(k)] \begin{bmatrix} 0 \\ \bar{w}_{yy}^{(o)} \\ \hline 0 \\ \bar{w}_{yu}^{(n)} \\ \hline 0 \\ \bar{w}_{uy}^{(o)} \\ \hline 0 \\ \bar{w}_{uu}^{(o)} \end{bmatrix}$$

or

$$\bar{w}_{yy}^{(o)} = 1, \bar{w}_{yu}^{(o)} = \bar{w}_{uy}^{(o)} = 0 \quad (A-22)$$

Substituting (A-21) and (A-22) into (A-20) yields

$$\begin{bmatrix} w_{yy}^{(1)} \\ \hline w_{yu}^{(1)} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \hline 0 \\ 0 \end{bmatrix} - \underline{K}^{(1)T} \begin{bmatrix} 0 \\ 1 \\ \hline 0 \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} 1 \\ 0 \\ \hline 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ k_{11}^{(1)} \\ \hline 0 \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ \hline k_{21}^{(1)} \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ -k_{11}^{(1)} \\ \hline 0 \\ -k_{21}^{(1)} \end{bmatrix}$$

(A-23)

This result can be substituted into (A-11)

$$\begin{aligned} \underline{e}^{(1)}(k) &= \begin{bmatrix} e_y^{(1)}(k) \\ e_u^{(1)}(k) \end{bmatrix} = [y(k)y(k-1) | u(k)u(k-1)] \begin{bmatrix} 1 \\ -k_{11}^{(1)} \\ 0 \\ -k_{21}^{(1)} \\ 0 \\ -k_{12}^{(1)} \\ 1 \\ -k_{22}^{(1)} \end{bmatrix} \\ &= \begin{bmatrix} y(k) - k_{11}^{(1)}y(k-1) - k_{21}^{(1)}u(k-1) \\ -k_{12}^{(1)}y(k-1) + u(k) - k_{22}^{(1)}u(k-1) \end{bmatrix} \end{aligned} \quad (A-24)$$

Inspection of Figure 2.8 shows that (A-24) is indeed the equation of the first stage error signal.

An analogous procedure can be allowed to develop the recursive solution for the backward weights. Substituting (A-12) and (A-17) into (A-4) yields

$$\underline{\bar{e}}^{(n+1)}(k) = [\underline{y}^{(n+1)T}(k) | \underline{u}^{(n+1)T}(k)] \begin{bmatrix} 0 \\ \underline{\bar{w}}_{yy}^{(n)} \\ 0 \\ \underline{\bar{w}}_{yu}^{(n)} \\ 0 \\ \underline{\bar{w}}_{uy}^{(n)} \\ 0 \\ \underline{\bar{w}}_{uu}^{(n)} \end{bmatrix} - \underline{K}^{(n+1)T} \begin{bmatrix} \underline{w}_{yy}^{(n)} \\ 0 \\ \underline{w}_{yu}^{(n)} \\ 0 \\ \underline{w}_{uy}^{(n)} \\ 0 \\ \underline{w}_{uu}^{(n)} \\ 0 \end{bmatrix} \quad (A-25)$$

Comparing with (A-17) and adjusting for order (n+1) reveals

$$\begin{bmatrix} \bar{w}_{yy}^{(n+1)} \\ \text{-----} \\ \bar{w}_{yu}^{(n+1)} \\ \text{-----} \\ \bar{w}_{uy}^{(n+1)} \\ \text{-----} \\ \bar{w}_{uu}^{(n+1)} \end{bmatrix} = \begin{bmatrix} 0 \\ \bar{w}_{yy}^{(n)} \\ \text{-----} \\ 0 \\ \bar{w}_{yu}^{(n)} \\ \text{-----} \\ 0 \\ \bar{w}_{uy}^{(n)} \\ \text{-----} \\ 0 \\ \bar{w}_{uu}^{(n)} \end{bmatrix} \bar{K}^{(n+1)T} \begin{bmatrix} w_{yy}^{(n)} \\ 0 \\ \text{-----} \\ w_{yu}^{(n)} \\ 0 \\ \text{-----} \\ w_{uy}^{(n)} \\ 0 \\ \text{-----} \\ w_{uu}^{(n)} \\ 0 \end{bmatrix} \quad (A-27)$$

With the weight vectors for all stages established, the reflection factors may be found in terms of these weights. Substituting (A-12) and (A-17) into (A-8) yields

$$\underline{\Delta}^{(n)} = \varepsilon \{ \underline{e}^{(n)}(k) \bar{e}^{(n)}(k-1) \} =$$

$$\underbrace{\begin{bmatrix} \underline{y}^{(n+1)T}(k) & \underline{u}^{(n+1)T}(k) \end{bmatrix}}_{\varepsilon} = \begin{bmatrix} \begin{bmatrix} \underline{w}_{yy}^{(n)} & 0 & \underline{\underline{w}}_{yu}^{(n)} & 0 \end{bmatrix} & \begin{bmatrix} 0 & \underline{\underline{w}}_{yy}^{(n)} & 0 & \underline{\underline{w}}_{yu}^{(n)} \end{bmatrix}^T \\ \begin{bmatrix} \underline{w}_{uy}^{(n)} & 0 & \underline{\underline{w}}_{uu}^{(n)} & 0 \end{bmatrix} & \begin{bmatrix} 0 & \underline{\underline{w}}_{uy}^{(n)} & 0 & \underline{\underline{w}}_{uu}^{(n)} \end{bmatrix} \end{bmatrix} \begin{bmatrix} \underline{y}^{(n+1)}(k) \\ \underline{\underline{u}}^{(n+1)}(k) \end{bmatrix}$$

$$\left\{ \begin{array}{l} \left[\bar{y}^{(n+1)T}(k) \mid \bar{u}^{(n+1)T}(k) \right] \\ \left[\bar{y}^{(n+1)}(k) \mid \bar{u}^{(n+1)}(k) \right] \end{array} \right\} \varepsilon \left\{ \begin{array}{l} \left[\begin{array}{c|c|c} \bar{w}_{yy}^{(n)} & [0 \bar{w}_{yy}^{(n)}]^T & \bar{w}_{yy}^{(n)} \\ 0 & | & 0 \\ \hline \bar{w}_{uy}^{(n)} & [0 \bar{w}_{uy}^{(n)}]^T & \bar{w}_{uy}^{(n)} \\ 0 & | & 0 \end{array} \right] & \left[\begin{array}{c|c|c} \bar{w}_{yy}^{(n)} & [0 \bar{w}_{yy}^{(n)}]^T & \bar{w}_{yy}^{(n)} \\ 0 & | & 0 \\ \hline \bar{w}_{uy}^{(n)} & [0 \bar{w}_{uy}^{(n)}]^T & \bar{w}_{uy}^{(n)} \\ 0 & | & 0 \end{array} \right] \\ \left[\begin{array}{c|c|c} \bar{w}_{yy}^{(n)} & [0 \bar{w}_{yy}^{(n)}]^T & \bar{w}_{yy}^{(n)} \\ 0 & | & 0 \\ \hline \bar{w}_{uy}^{(n)} & [0 \bar{w}_{uy}^{(n)}]^T & \bar{w}_{uy}^{(n)} \\ 0 & | & 0 \end{array} \right] & \left[\begin{array}{c|c|c} \bar{w}_{yy}^{(n)} & [0 \bar{w}_{yy}^{(n)}]^T & \bar{w}_{yy}^{(n)} \\ 0 & | & 0 \\ \hline \bar{w}_{uy}^{(n)} & [0 \bar{w}_{uy}^{(n)}]^T & \bar{w}_{uy}^{(n)} \\ 0 & | & 0 \end{array} \right] \end{array} \right\}$$

The center matrix is dimensioned (2×2) . Its four elements are each a submatrix of dimension $(2(n+2) \times 2(n+2))$. Each submatrix is premultiplied by a $2(n+2)$ element row vector and postmultiplied by a $2(n+2)$ element column vector. The result of these multiplications will be a scalar value for each of the four elements of $\underline{\Delta}^{(n)}$.

The order of these vector multiplications may be inverted yielding

$\Delta^{(n)} =$

$$\begin{bmatrix} [w_{yy}^{(n)}]^T \begin{vmatrix} (n)^T & 0 \\ 0 & w_{yu}^{(n)} \end{vmatrix} & [y^{(n+1)}(k)] \begin{vmatrix} (n+1)^T & 0 \\ 0 & u^{(n+1)}(k) \end{vmatrix} & [y^{(n+1)}(k)] \begin{vmatrix} (n+1)^T & 0 \\ 0 & u^{(n+1)}(k) \end{vmatrix} & [y^{(n+1)}(k)] \begin{vmatrix} (n+1)^T & 0 \\ 0 & u^{(n+1)}(k) \end{vmatrix} \\ \vdots & \vdots & \vdots & \vdots \\ [w_{yy}^{(n)}]^T \begin{vmatrix} (n)^T & 0 \\ 0 & w_{yu}^{(n)} \end{vmatrix} & [y^{(n+1)}(k)] \begin{vmatrix} (n+1)^T & 0 \\ 0 & u^{(n+1)}(k) \end{vmatrix} & [y^{(n+1)}(k)] \begin{vmatrix} (n+1)^T & 0 \\ 0 & u^{(n+1)}(k) \end{vmatrix} & [y^{(n+1)}(k)] \begin{vmatrix} (n+1)^T & 0 \\ 0 & u^{(n+1)}(k) \end{vmatrix} \end{bmatrix}$$

$$= \begin{bmatrix} \Delta_{11} & \Delta_{12} \\ \Delta_{21} & \Delta_{22} \end{bmatrix}$$

(A-31)

Extracting Δ_{11} and rewriting

$$\Delta_{11} = \begin{bmatrix} \underline{w}_{yy}^{(n)T} & 0 \\ 0 & \underline{w}_{yu}^{(n)T} \end{bmatrix} \epsilon \left\{ \begin{bmatrix} \underline{y}^{(n+1)}(k) \\ \text{-----} \\ \underline{u}^{(n+1)}(k) \end{bmatrix} \begin{bmatrix} \underline{y}^{(n+1)T}(k) & \underline{u}^{(n+1)T}(k) \end{bmatrix} \right\} \begin{bmatrix} 0 \\ \underline{w}_{yy}^{(n)} \\ \text{-----} \\ 0 \\ \underline{w}_{yu}^{(n)} \end{bmatrix}$$

$$= \begin{bmatrix} \underline{w}_{yy}^{(n)T} & 0 \\ 0 & \underline{w}_{yu}^{(n)T} \end{bmatrix} \begin{bmatrix} \underline{R}_{yy}^{(n)} & \underline{R}_{yu}^{(n)} \\ \text{-----} & \text{-----} \\ \underline{R}_{uy}^{(n)} & \underline{R}_{uu}^{(n)} \end{bmatrix} \begin{bmatrix} 0 \\ \underline{w}_{yy}^{(n)} \\ \text{-----} \\ 0 \\ \underline{w}_{yu}^{(n)} \end{bmatrix} \quad (\text{A-32})$$

where

$$\underline{R}_{xz}^{(n)} = \begin{pmatrix} R_{xz}(0) & \dots & R_{xz}(-n-1) \\ \vdots & & \vdots \\ R_{xz}(n+1) & \dots & R_{xz}(0) \end{pmatrix} \quad (\text{A-33})$$

is a $(n+2) \times (n+2)$ (cross) correlation array.

By defining a $2(n+2) \times 2(n+2)$ matrix $\underline{R}^{(n)}$ as

$$\underline{R}^{(n)} = \left[\begin{array}{c|c} \underline{R}_{yy}^{(n)} & \underline{R}_{yu}^{(n)} \\ \hline \underline{R}_{uy}^{(n)} & \underline{R}_{uu}^{(n)} \end{array} \right] \quad (\text{A-34})$$

the elements of the matrix $\Delta^{(n)}$ may be defined

$$\Delta_{11}^{(n)} = \left[\underline{w}_{yy}^{(n)T} \mid 0 \mid \underline{w}_{yu}^{(n)T} \mid 0 \right] \underline{R}^{(n)} \left[\begin{array}{c} 0 \\ \underline{w}_{yy}^{(n)} \\ \hline 0 \\ \underline{w}_{yu}^{(n)} \end{array} \right] \quad (\text{A-35})$$

$$\Delta_{12}^{(n)} = \left[\underline{w}_{yy}^{(n)T} \mid 0 \mid \underline{w}_{yu}^{(n)T} \mid 0 \right] \underline{R}^{(n)} \left[\begin{array}{c} 0 \\ \underline{w}_{uy}^{(n)} \\ \hline 0 \\ \underline{w}_{uu}^{(n)} \end{array} \right] \quad (\text{A-36})$$

$$\Delta_{21} = \begin{bmatrix} \underline{w}_{uy}^{(n)T} & 0 & \underline{w}_{uu}^{(n)T} & 0 \end{bmatrix} \underline{R}^{(n)} \begin{bmatrix} 0 \\ \underline{w}_{yy}^{(n)} \\ \text{-----} \\ 0 \\ \underline{w}_{yu}^{(n)} \end{bmatrix} \quad (\text{A-37})$$

$$\Delta_{22} = \begin{bmatrix} \underline{w}_{uy}^{(n)T} & 0 & \underline{w}_{uu}^{(n)T} & 0 \end{bmatrix} \underline{R}^{(n)} \begin{bmatrix} 0 \\ \underline{w}_{uy}^{(n)} \\ \text{-----} \\ 0 \\ \underline{w}_{uu}^{(n)} \end{bmatrix} \quad (\text{A-38})$$

Implementation is discussed in Chapter IV.B.

LIST OF REFERENCES

1. Burg, J.P., Maximum Entropy Spectral Analysis, PhD Dissertation, Stanford University, May 1975.
2. Durbin, J., "The Fitting of Time Series Models", Rev. Inst. Int. Statist., Vol. 28, No. 3, pp. 233-243, 1960.
3. Griffiths, L.S., "A Continuously Adaptive Filter Implemented as a Lattice Structure", Proc. IEEE ICASSP, (Hartford CT), pp. 683-686, May 1977.
4. Itakura, F. and Saito, S., "On the Optimum Quantization of Feature Parameters in the Parcor Speech Synthesizer", Proc. 1972 Conf Speech Commun. Process, pp. 434-437, 1972.
5. Kalman, R.E., "Design of a Self-Optimizing Control System", Trans. A.S.M.E., Vol. 80, pp. 468-478, Feb. 1958.
6. Levinson, N., "The Weiner RMS (root mean square) Error Criterion in Filter Design and Prediction", J. Math Phys., Vol. 25, No. 4, pp. 261-278, 1947.
7. Nuttall, A.H., "Multivariate Linear Predictive Spectral Analysis Employing Forward and Backward Averaging: A Generalization of Burg's Algorithm", NUSC Technical Report 5501, New London, CT, 13 Oct 1976.
8. Makhoul, J., "A Class of All Zero Lattice Digital Filters: Properties and Applications", IEEE Trans on A.S.S.P., Vol. A.S.S.P. 26, No. 4, pp. 304-314, Aug 1978.
9. Makhoul, J., "Linear Prediction: A Tutorial Review", Proc. IEEE, Vol. 63, No. 4, pp. 561-580, Apr 1975.
10. Makhoul, J., "Stable and Efficient Methods for Linear Prediction", IEEE Trans on A.S.S.P., Vol A.S.S.P. 25, No. 5, pp. 423-428, Oct 1977.
11. Markel, J.D. and Gray, A.H., Jr., Linear Prediction of Speech, pp. 20-41, Springer Verlag, 1975.
12. Mendel, J.M., Discrete Techniques of Parameter Estimation, Marcel Dekker Inc., 1973.
13. Morf, M., et al, "Recursive Least Squares Ladder Estimation Algorithms", IEEE Trans. on A.S.S.P., Vol. A.S.S.P. 29, No. 3, June 1981.

14. Oppenheim, A.V., and Schafer, R.W., Digital Signal Processing, Prentice Hall, 1975.
15. Perry, F.A., Parametric Modeling of Linear and Nonlinear Systems, PhD Dissertation, U.S. Naval Postgraduate School, June, 1980.
16. Robinson, E.A., Multichannel Time Series Analysis with Digital Computer Programs, Rev-Ed., Holden-Day, 1967.
17. Weiner, N., The Extrapolation, Interpolation, and Smoothing of Stationary Time Series with Engineering Applications, Wiley, New York, 1949.
18. Widrow, B., et al, "Adaptive Antenna Systems", Proc. IEEE, Vol. 55, pp. 2143-2159, Dec 1967.
19. Yule, G.U., "On a Method of Investigating Periodicities in Disturbed Series, with Special Reference to Wolfer's Sunspot Numbers", Phil. Trans. Roy Soc., Vol. 226-A, pp. 267-298, 1927.

BIBLIOGRAPHY

Brigham, E.O., The Fast Fourier Transform, Prentice Hall, Inc., 1974.

Gray, A.H., and Markel, J.D., "A Spectral Flatness Measure for Studying the Autocorrelation Method of Linear Prediction of Speech Analysis", IEEE Trans. on A.S.S.P., Vol. A.S.S.P. 22, No. 3, June 1974, pp. 207-217.

Kay, S.M., and Marple, S.L., "Spectrum Analysis - A Modern Perspective", Proc IEEE, Vol. 69, No. 11, Nov 1981, pp. 1380-1419.

Makhoul, J.I. and Wolf, J.J., Linear Prediction and the Spectral Analysis of Speech, Bolt Beranek and Newman Inc., BBN Report No. 2304, 31 Aug 1972.

Makhoul, J., "A Class of All Zero Lattice Methods for Linear Prediction", IEEE Trans. on A.S.S.P., Vol. A.S.S.P. 26, No. 4, Aug 1978, pp. 304-314.

Papoulis, A., "Maximum Entropy and Spectral Estimation: A Review", IEEE Trans. on A.S.S.P., Vol. A.S.S.P. 29, No. 6, Dec 1981, pp. 1176-1186.

Rabiner, R.R., and Schafer, R.W., Digital Processing of Speech Signals, Prentice Hall, Inc.

Romeo, M., Multichannel ARMA Lattice Modeling, With Applications to the Phase Locked Loop, Engineers Thesis, Naval Postgraduate School, Dec 1980.

Tuttle, D.F., Electric Networks Analysis and Synthesis, McGraw-Hill, 1965.

INITIAL DISTRIBUTION LIST

	No. Copies
1. Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2
2. Library, Code 0142 Naval Postgraduate School Monterey, California 93940	2
3. Department Chairman, Code 62 Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
4. Professor S. R. Parker, Code 62Px Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	10
5. Professor K. G. Gray, Code 62Gy Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
6. Professor G. A. Myers, Code 62Mv Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	-
7. Professor U. R. Kodres, Code 52Kr Department of Computer Science Naval Postgraduate School Monterey, California 93940	1
8. Professor B. O. Shubert, Code 55Sy Department of Operations Research Naval Postgraduate School Monterey, California 93940	1
9. Lieutenant Commander D. J. Klich USS EDWARD MCDONNEL (FF 1043) FPO, Miami, FL 34092	5
10. Lieutenant Commander Ihn-Kiel Chang 665-1-4 Yuk Sam-Dong Kang Nam-Gu, Seoul, Korea	1